Third Edition

The Handbook of Personal Computer Instrumentation

for Data Acquisition, Test Measurement and Control

BURR-BROWN®

Intelligent Instrumentation Products

MODEL NUMBER INDEX

(Subject Index at back of handbook)

PCI-20000 System

Other PCI Products

BURR-BROWN CORPORATION
INTELLIGENT INSTRUMENTATION PRODUCTS

1141 W. GRANT ROAD #131 • TUCSON, AZ 85705
TELEFAX: 602-623-8965 • TWX: 910-997-0735 • TELEX: 88-8225

TELEPHONE INFORMATION

To place an order, or for information about products, pricing, and delivery, USA customers please call a local representative (office listing at back of this handbook) or call Burr-Brown in Tucson, AZ (602) 746-1111.

For in-depth applications information, call (602)-624-2434.

International Customers please call a local office (listing at back of this handbook).

LI-334B Printed in U.S.A., January 1988

BURR-BROWN®

PREFACE TO THE THIRD EDITION

THE UBIQUITOUS COMPUTER

Today the personal computer is everywhere, a powerful tool for industry and scientific research.

Teamed with a precision analog/digital interface system, a personal computer can effectively perform all the functions of larger, dedicated instrumentation and control systems —and at a fraction of their former cost. And with many software and hardware options currently available, the personal computer interface can be easily configured and expanded to satisfy a wide variety of instrumentation and control applications.

THE RIGHT TOOL SET

Personal computers and their software provide a standard set of tools for data management tasks. New systems and software are evolving for more specialized applications in industry—special tools tailored for use in the areas of:

- data acquisition
- test instrumentation
- measurement
- control

These new tools make the power, versatility and low cost of the personal computer available for a wide range of new and demanding applications.

Now an engineer can combine the tools to acquire data, test products and systems, and perform controls using these same tools to organize, display, provide graphics, and print data/reports. Powerful systems are thus flexible, programmable and inexpensive.

NEED FOR A HANDBOOK

The rapid development, evolution, and acceptance of the personal computer along with personal-computer instrumentation (PCI) systems, such as the PCI-20000, have created a need for a handbook to better acquaint engineers with just what these newest tools really are. The handbook explains what an engineer should consider as he begins to configure PCI systems and to use them to increase his productivity.

So here it is, a **real** handbook (and not just a set of product data sheets). This handbook begins with tutorial information to help a new user to get started in the world of data acquisition. The tutorial section is followed by a large applications section, including many specific applications examples and sample programs. At the end, there are specifications, technical information, and how-to-configure charts for the PCI products that are the leadership products for the personal computer instrumentation industry today.

THIRD EDITION

An extremely large, worldwide demand for the first and second editions of the PCI Handbook has brought about the availability of the third edition sooner than expected. This third edition contains additional pertinent information in each one of the major categories: tutorial, applications, and product specifications. This handbook is already becoming the industry's standard-of-reference for Personal Computer Instrumentation used in data acquisition, test, measurement, and control.

> Free Burr-Brown Demonstration Diskettes showing product capabilities, specifications, and applications for the PCI-20000 system are available through Burr-Brown sales offices. These diskettes run on the IBM PC and compatible computers containing a graphics card. Please contact your local sales office for your free diskette. See office listings at the back of this handbook.

TAB INDEX

Section 1
Data Acquisition and Control, An Overview

The term data acquisition and control (DA&C) can mean different things to each of us. Observing a voltmeter and manually recording its reading certainly constitutes data acquisition. Furthermore, when we turn the dimmer knob on a room lamp, we achieve control. These are, of course, very simple examples. Think about the number of DA&C actions taken by each of us at work, in industry, in driving a car, in everyday life.

Simply stated, data acquisition is the collecting of information that describes a given situation. The data typically reflects what was happening when a given condition was satisfied. Usually, this condition is defined by a uniform time-base, but it could be controlled by any event. "Real-time" systems are characterized by their ability to perform a given data acquisition and/or control task within an appropriate time window. How fast such a system must respond depends upon the speed and accuracy requirements of that given application.

For every data acquisition or control system, no matter how slow it may be, there is an application sufficiently slow that, for that application, the system is real-time.

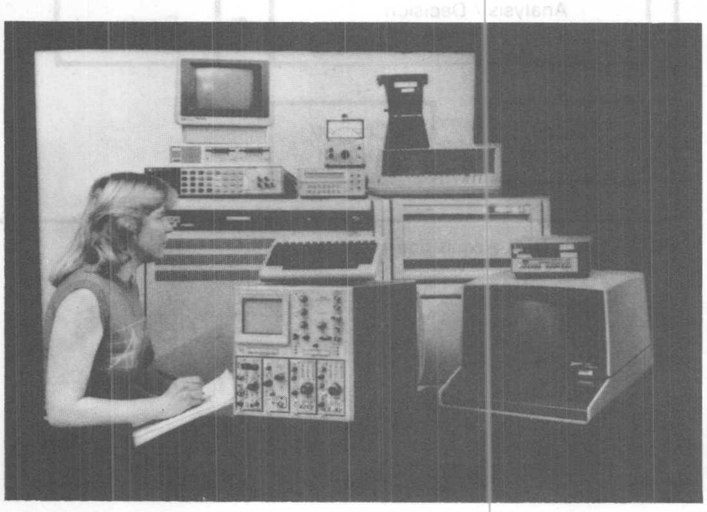
Control implies the generation of an output signal in response to input data. Control can be "open loop" or "closed loop." Turning off the heat at 4:00 PM is an example of open-loop control, while turning off the heat because it is too hot represents closed-loop control.

Data is collected by technicians, engineers, physicists, chemists or others involved in research, test, development, production, quality control, management, process control, etc.. Industries involved include: electrical, electronics, steel, mechanical, chemical, oil, food, energy, genetics, medical, and paper. Data can be collected by anyone, anywhere to deduce trends, establish alarms, make decisions, and control operations.

We have been data takers since the beginning of time. We have sensed our environment and learned to take beneficial action. We read thermometers, voltmeters, scales and oscilloscopes. We record the data, analyze it, use it, and communicate it. However, methods are changing. Now the emphasis is on getting machines to meet many of our data acquisition and control needs. The unselfish motive is productivity. Speed, accuracy, dependability, reliability and cost are related factors.

In the past when process monitoring was the principal task, an automatic data logger was the accepted form of automation. Data loggers include strip chart recorders, printers, and tape recorders. When monitoring alone was not enough, programmable controllers were often matched to the requirements of the job. However, in an increasing number of applications, data loggers and programmable controllers could not do everything that was desired. This was due, in part, to the narrow range of functions supported by their hardware and software. In contrast, today's state-of-the-art systems, offering a full range of capabilities, are based upon our most effective productivity machine, the modern digital computer.

Figure 1.1 suggests the components of a data acquisition



Data Acquisition Systems Take Many Forms, Including Pencil and Paper.

and control system. The computer not only provides the analysis and decision-making capability, but also controls the active signal-conditioning and data-conversion functions. A given system might not include all of the elements shown in Figure I.1. In this handbook, all further references to data acquisition or control apply to those applications in which a computer plays an important role.

Modern computers offer high speed, flexibility, adaptability, consistency, reliability and mass memory. These features provide extensive capabilities for mathematics, analysis, storage, display, report generation, control and communications. However, most real-world signals (temperature, pressure, flow, speed, intensity, position, etc.) cannot be read directly by digital computers. These parameters are represented by analog signals distinguished by their continuum of levels. However, computers can recognize only digital (off or on) levels. Therefore, a translation-type product is required.

The Link—Data acquisition and control products translate real-world signals into a format that digital computers can accept. DA&C systems can also regenerate analog and other signals from computer instructions. In this way DA&C systems bridge the gap between the pervasive digital computer and the real world.

The personal computer (PC), in contrast to other forms of computers, is the fastest growing "engine" for new DA&C system designs. Many of the reasons for this trend will be explored in the next section.

The PC has already made significant inroads into many important applications areas. These include:

- Laboratory data collection and automation
- Medical instrumentation and patient monitoring
- Automatic test equipment (ATE) for incoming inspection, life test, burn in, production test, and final test
- Industrial monitoring and control
- Environment and utility management.

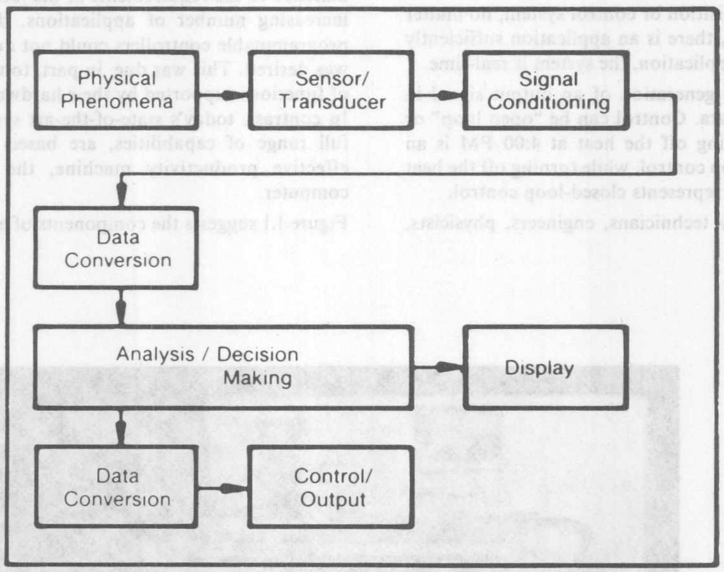


FIGURE I.1. Data Acquisition and Control Flow Diagram.

Section 2
Different Types Of Systems & How They Connect To The Personal Computer

An important part of any data acquisition system is its host computer. There are two possible ways for the DA&C hardware to interface with the computer: direct connection to the PC bus (internal bus products) or connection via a standard communication channel such as RS-232, RS-422 or IEEE-488 (external bus products). Each method has its advantages and disadvantages.

Throughout this section, the term "system" will be used in several ways. A system can include everything that is required to perform the complete data acquisition task, including the host computer. The term "system" can also be used to describe a group of circuit elements. Perhaps this should more accurately be called a subsystem.

External Bus Products - The advantages associated with external bus products (using RS-232, etc.) include:

- Virtually any size system can be configured
- The DA&C system can be placed remotely from the host computer (so that the DA&C system can be located close to the field signals)
- The DA&C system has the possibility of off-loading some of the data-collecting tasks from the host computer
- The DA&C system can be interfaced to virtually any type of computer

Figure 2.1 shows a simplified block diagram of an external bus system. Communications through RS-232, RS-422 or IEEE-488 requires the data acquisition system to have its own internal microprocessor. This local microcomputer also facilitates remote operation and helps achieve the reduced load on the host PC. Systems of this type reside in their own enclosures. These enclosures or "boxes" provide space not only for the microcomputer, but also for the power supplies and the analog and digital input/output hardware. In most cases the I/O functions are grouped by type on individual plug-in boards. This allows both the selection of I/O types and the selection of the number of channels to be supported. To facilitate very large point-count systems, add-on expansion enclosures are also available.

The ability to have remote (distant from the host computer) DA&C boxes allows the construction of distributed systems. Thus, a large number of parameters can be monitored or controlled even though they physically originate far from each other and far from the host PC. For example, the data from 31 different production lines, each with separate DA&C subsystems, can be interconnected via RS-422. This allows monitoring by a single PC, which could be in a supervisor's office, in another building several thousand feet (or hundreds of meters) away. This type of capability can greatly improve productivity and reduce overall system cost.

Internal Bus Products - The main advantages of making direct connection to the PC bus include:

- High speed
- Low-cost
- Smaller size

Cost is reduced with this kind of DA&C system because it does not require its own separate enclosure or power supply. Power is obtained from the PC. When the data acquisition hardware resides inside the host computer, important advantages in both size and space utilization are obtained. High speed is achieved by eliminating the relatively slow, external, communications-channel protocol. As an example, the data acquisition rate using RS-232 at 9600 baud is limited to about 20 analog channels per second. In contrast, some direct PC bus products can take data faster than 100,000 channels per second. Figure 2.2 shows a simplified block diagram of an internal bus system.

Two major types of internal bus (PC bus) products exist, distinguished by the way in which the Input/Output

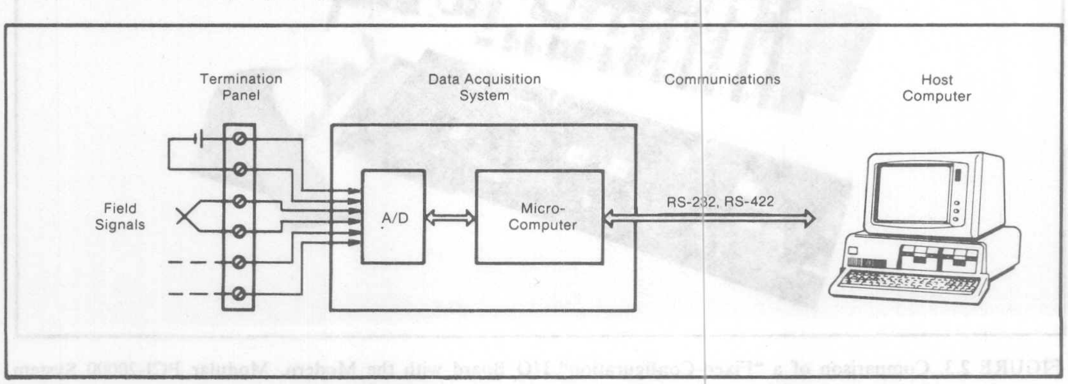


FIGURE 2.1. An External Bus DA&C System, Block Diagram.

2
DATA ACQUISITION SYSTEMS
CONNECTION TO THE PERSONAL COMPUTER

channels are configured. Both are board-level systems that make direct connection to the computer expansion bus, yielding the speed and cost advantages mentioned above. Some boards have a "fixed" arrangement of analog and digital input/outputs. This means that whatever configuration one buys, that's what he has regardless of future needs. Limitations of this type of system include lack of channel expansion capability, and the inability to add functions not originally purchased. In contrast, second-generation products, which are modular in nature, allow the user to select, even in the field, the quantity and configuration of the I/O functions desired. This feature is provided by a family of function (or instrument) modules. Thus, the more modern, modular, board-level systems share some of the positive features of the box systems. These include expandability and user selection of I/O functions.

In most applications, fixed-board I/O configurations require significant compromise. Either the number of channels desired cannot be obtained, or the user must purchase functions not required. With the great diversity of uses, it is inevitable that a mismatch between the available I/O and the actual requirements will exist. Some fixed-configuration products allow for selected types of channel expansion via external add-on boards or boxes. When cost, space, and ease of use are considered, this type of product is less attractive than systems that can meet all I/O requirements inside the host computer. Modular board-level systems are far more effective in this regard and are readily tailored for specific applications. The PCI-20000 system represents the state-of-the-art in modular plug-in board systems for the PC-bus. Detailed information and specifications on the PCI-20000 system are included in later sections of this handbook.

Figure 2.3 is a photograph showing a "Fixed-configuration" or "combination" I/O board alongside the PCI-20000 System, which is a new, modern, modular type of I/O product.

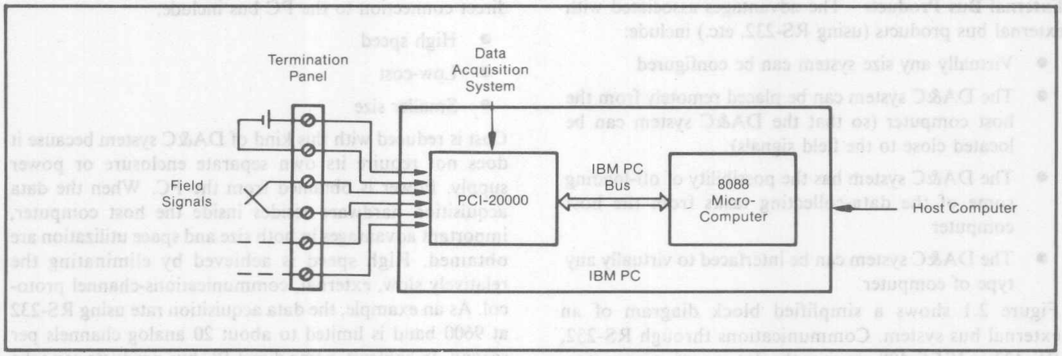


FIGURE 2.2 Internal PC Bus System.

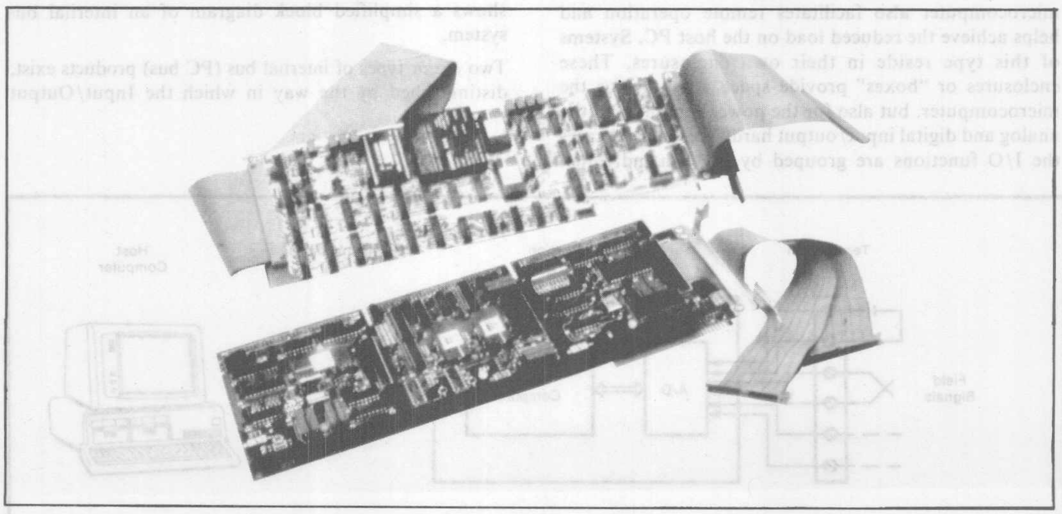


FIGURE 2.3. Comparison of a "Fixed-Configuration" I/O Board with the Modern, Modular PCI-20000 System (top board is a fixed configuration; the lower boards are PCI-20000).

Section 3
Personal Computers in Data Acquisition & Control

Historically, industrial and scientific data acquisition and control (DA&C) tasks were implemented with large mainframe or minicomputer systems. Typically, these were powerful 16-bit machines that ran in time-sharing or multitasking modes. Their complexity and expense dictated that they be configured as centralized utilities shared by many users and applications. Small or remote jobs were often relegated to manual, or at best, simple electronic data-logging techniques. These tasks could not justify the capital expense or manpower overhead of computerization. Thus, these smaller tasks could not benefit from the flexibility and power of a computerized solution.

The advent of the modern personal computer (PC) makes it possible for virtually everyone to take advantage of the flexibility, power and efficiency of computerized data acquisition and control. PCs offer high performance and low cost along with an ease of use that is unprecedented. Thanks to a significant degree of standardization among PC and DA&C manufacturers, a large family of hardware/software tools and applications packages have evolved. The result is that an individual engineer or scientist can now implement a custom DA&C system within a fraction of the time and expense formerly required. It is now practical to tailor highly efficient solutions to unique applications. Furthermore, personal computers invite innovation. This type of innovation has revolutionized the office and is now revolutionizing factories, production lines, testing, and laboratories. Figure 3.1 shows the relationship between the DA&C system and its host computer.

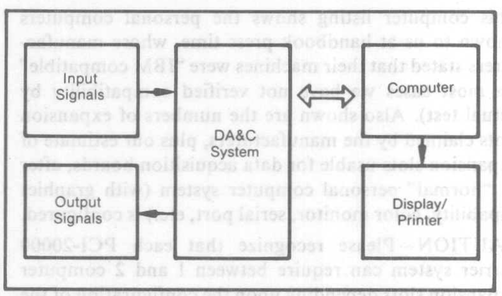


FIGURE 3.1. A Computerized Data Acquisition & Control System.

Because of the obvious advantages of personal computers, along with their increasingly widespread usage, we will assume the use of a PC in the remainder of this handbook. However, it is important to remember that many of the systems described can be used with virtually any type of computer.

There are several popular architectures associated with today's PCs. In general, most computer families are differentiated by the microcomputer chip used. For example: 8080, 8088, 8086, 80286 (from Intel), 68000 (from Motorola) and Z80, Z8000 (from Zilog). More about the differences in some of these microprocessors will be described later. Each type is enjoying (or has enjoyed) a strong market share in specific areas (i.e., home, office, etc.). Recent studies indicate that technical, scientific and industrial applications are dominated by computers conforming to the de facto standards set down by IBM.

The original IBM PCs were built around the 8088, while recent offerings utilize the 80286. The free-enterprise system has yielded a vast array of IBM PC "work-alikes". Therefore, when we refer to the IBM PC (or just PC), we are referring to this generic family of compatible computers. Some of these competing models offer advantages in features, performance and/or cost. A partial listing of IBM-compatible computers is shown later in this section for reference purposes.

In the context of DA&C applications, "true" IBM PC compatibility includes both hardware and software requirements. Only those computers that can run, without modification, the same software written for the IBM PC are compatible. Likewise, a compatible machine must accept the same range of add-on (or add-in) boards that plug directly inside the IBM PC.

INSIDE THE IBM PC

The PC consists of a system unit (microcomputer, memory, power supply, etc.), a keyboard and one or more output devices such as monitors, printers and plotters. The system unit is usually housed in an enclosure, separate from other major components such as the monitor. An exception to this rule would be portable PCs, which integrate all devices into an easy-to-carry case (example: Compaq portables).

"Lap-top" and other miniature computers are usually not included in the true PC-compatible category, because they lack expansion slots. This definition could be debated because some small computers do offer external expansion capabilities.

An expansion slot is a physical and electrical space set aside for the connection of accessory hardware items to the PC. Electrical connection is made directly to the internal microcomputer bus. These accessory items usually take the form of a plug-in, printed-circuit board (i.e., a graphics interface, a memory expansion module or a data acquisition device).

Plug-in boards can be designed to be addressed by the microcomputer in two different ways, either as I/O ports or as memory locations. There are advantages to both systems. However, memory addressing offers a higher level of performance that includes improved speed, extended address space, and the full use of the processor's instruction set.

Some of the computer I/O and memory addresses are reserved by the computer manufacturers for standard

functions (i.e., graphics cards, RS-232 ports, memory, disk controllers, etc.). Most other types of plug-in boards are equipped with a bank of switches that allow the user to select an appropriate address location. Included at the end of this handbook section is a "map" that identifies both the I/O and the memory-address allocations for the most common PCs. Thus, if you know your computer's hardware configuration, using the map will make the selection of additional available address locations easy.

PCs employ several distinct memory types: RAM, ROM, floppy disk and hard disk. Other memory technologies include magnetic tape and bubble. RAM and ROM are semiconductor devices offering very high-speed operation. Random-access memory (RAM) has both read and write capabilities that can be accessed by the microcomputer. Read-only memory (ROM), on the other hand, contains a fixed set of information that can only be read by the microcomputer. The microcomputer itself is often referred to as the central processing unit (CPU).

The name RAM is somewhat of a misnomer, because both RAM and ROM along with many other memory types are random-access. ROM is pre-programmed at the factory to contain the most fundamental CPU operating instructions. This includes the code required to start or to "boot" the computer, and is known as the BIOS (basic input/output system). All other active program information is contained in RAM. The amount of RAM that can be used is determined by the particular microcomputer chip used and the available software. The 20-bit address bus of the 8088 limits memory locations to about 1Mbyte. The 80286 has a 24-bit bus which can address about 16Mbytes. RAM is normally termed "volatile", because in most systems its data will be destroyed if power is lost. Permanent storage of data and programs is usually provided by the disk drives.

Most people who use computers will not wish to "talk" directly with the CPU, BIOS or the disk drives because of the complexities involved. Extensive interface software has been developed to bring the power of the PC within easy reach of non-specialists. This software is known as the operating system (OS). The most widely used OS is PC or MS DOS (disk operating system). Other operating systems include CP/M and Unix.

Even among compatible PCs, the effectiveness of the expansion slots varies considerably. The main considerations include: the number of available slots, spacing between slots, length of the slots and the power supply available. Because of their mechanical design, some computer types make it easier than others to insert boards into expansion slots.

The speed at which a PC can process instructions (i.e., run a program) is dependent upon many factors. Some ingredients are under the programmer's control. Choice of language and the efficiency of the resulting code are significant. Software efficiency refers, in part, to how many machine cycles are required to execute the desired instructions. A "tight" program requires the fewest number of machine cycles and is thus very efficient.

Other factors are related to the PC's electrical design. Selection of the microcomputer chip, additional logic, circuit configuration and clock frequency are all important.

The clock frequency sets the speed of the microcomputer, but not necessarily its execution efficiency. For a given clock frequency one microcomputer can do more work than another. The 8088 and 80286 are both 16-bit chips. However, the 8088's bus can transfer only 8 bits at a time, compared to 16 bits for the 80286. If all other factors are equal, this results in a 2:1 speed advantage. Normally, the 8088 is clocked at 4.77MHz. 80286 machines are available with speeds of 6 to 12MHz. Still other characteristics of the 80286 contribute to improved efficiency.

In contrast to the many benefits of the 80286 machines, there is at least one important drawback. In presently available 80286 computers, the configuration of the direct memory access (DMA) circuitry is not optimized for DA&C. This can result in DMA transfers which are slower than expected (by as much as 2:1).

Computers are now starting to emerge that use the new 80386 processor. This chip promises to further increase speed and the amount of addressable RAM. Estimates are that 1 to 2 years will be required for vendors to produce the required software to exploit the 80386's full potential.

FIGURE 3.2 "Compatible" Computers.

This computer listing shows the personal computers known to us at handbook press time, whose manufacturers stated that their machines were "IBM compatible" (in most cases we have not verified compatibility by actual test). Also shown are the numbers of expansion slots claimed by the manufacturers, plus our estimate of expansion slots usable for data acquisition boards, after a "normal" personal computer system (with graphics capability, color monitor, serial port, etc.) is configured.

CAUTION—Please recognize that each PCI-20000 carrier system can require between 1 and 2 computer expansion slots depending upon the configuration of the I/O. In addition, it had been reported that some PC "clone" manufacturers have not implemented their DMA circuitry in the same way as IBM. The result is that the speed of DMA execution can be significantly reduced from normal specifications in these machines.

Manufacturer	Model Number	Processor	Estim. Total Expan. Slots	Usable Expan. Slots
Applied Digital Data	PC/1		5	2
Adept	PC	8088	6	2
Advanced Computer Prod.	Advanced 286	80286	8	4
Advanced Logic Research	Access 386	80386	8	4
	Advanced XT Turbo	8088	8	4
	Ideal XT Turbo	8088	8	4
Alphanumeric International	ANI-PC2	8088	8	4
	ANI-PCTurbo	8088	8	4
	ANIAT	80286	8	4
Amdex	RPC-50		11	6
American Computer & Perip.	286-A	80286	8	4
	88	8088	8	4
American Micro Tech.	AMT-286	80286	8	4
	AT jr.	8088-2	8	4
Anderson-Jacobson	AJ MBA		5	2
	AJ PHD		5	2
Apparat	Apparat Turbo PC	8088	8	4
AST	Premium/286	80286	7	4
AT&T	6300	8088	8	5
	6300 Plus	80286	8	5
ATronics Intl.	ATI System 286-12	80286	8	4
Bentley Computer Products	Bently	8088	8	4
	Bentley "T"	8088	8	4
	Bentley Turbo	8088	8	4
	Bentley XT	8088	8	4
	Bentley-286	80286	8	4
Bondwell Industrial Co.	30 Series	8088	5	2
CAD	Counsel Protean		8	
	Counsel Protean Plus		8	4
Canon	A-200		5	2
	A-200 HD		5	2
	A-200 II		5	2
Club AT	Club AT	80286	8	4
CMO	PC TOO	8088	8	4
Colby Research	PC 3		4	2
Columbia	1600-1	8088	8	4
	1600-4	8088	8	4
	MPC 4220	8088	8	4
	MPC 4620	8088	6	3
	MPC 4740	8088	5	2
	MPC 4820	8088	6	2
	MPC 4950 8088	5	2	
Commodore	PC-10	8088	4	2
	PC-20		3	1
Compaq	Deskpro	8086	8	4
	Deskpro 286	80286	8	4
	Deskpro 386	80386	4	2
	Plus	8088	3	2
	Portable	8088	5	2
	Portable II	80286	2	2
Computer Classifieds	CCI ST/286-12	80286	8	4
Computer Devices	DOT-2		2	2
Computer Dynamics	Mach 3	80286	8	4
Computerland	BC-286	80286	8	4
	BC-88	8088-2	8	4
Conroy-LaPointe	XT/1			2
Corona	ATP		5	2
	PC-11		4	2
	PC-400-22		4	2
	PCHD1		4	2
	Portable		4	2
Data General	Dasher/One2	8088	3	1
Datafox	286	80286	8	4
	88-2	8088-2	5	2
Digigraphic	70		7	3
Direct	IPC 1000		5	2
	PIC 1000 XD		5	2
Docutel-Olivetti	M18-PC	8088	4	2
	N24-PC	80286	7	3
Eagle	PC Plus 1	8088	4	2
	PC Plus 2	8088	4	2
	Plus XL		4	2
	Spirit 2		4	2
	Spirit XL		4	2
	Turbo GT/XI		8	4
Electro Design	Imp 12		12	8
	Imp 18		18	12
	Imp 6	8088	6	3
	Imp 8		8	4
	Imp-X	8088	12	8
Eltech Research	Jet-286	80286	8	4
Epson	Equity I	8088	3	1
	Equity II	8088	4	2
Ericsson	PC		6	2
Five Star Electronics	Five Star AT	80286	12	8
Gulfstream Microsystems	GMS PC/88-2	8088-2	8	4
	Professional 88	8088	8	4
Heath Co.	HF-148-1	8088		2
	HF-158-1	8088	6	2
	HF-158-41	8088	6	2
Hewlett-Packard	Vectra	80286	7	3
Hexace	System II		8	4
HiTech Intl.	SAM3001	80286	8	4
Honeywell	AP	80286		4
	XP	8088		4
IBM	5531	8088	8	4
	7531	80286	8	4
	7532	80286	8	4
	AT	80286	8	4
	PC	8088	5	2
	Portable	8088	5	1
	XT	8088	8	3
	XT, 286	80286	8	2
Intelligent Micro Systems	IMS-286	80286		2
ISI	6170		8	4
ITT	Xtra XP	80286		2
	Xtra1	8088	5	2
	Xtra2	8088	5	2
	Xtra3	8088	5	2
Jabberwauke	Turbo	8088	7	3
JC Information Systems	Lips III/286	NEC V40	8	4
Kaypro	286i	80286	8	4
	PC	8088	6	2
	PC 10	8088	6	2
Leading Edge	PC Mod D with HD	8088	3	1
	PC Model D	8088	4	2
	PC-1		7	3
	PC-5		7	3

Logic	AT	80286	8	4
	Turbo XT	8088	8	4
Logical Business Mach.	L-XT		3	1
Mad Computer	MAD-1		2	1
Maxum	286	80286	8	4
	286 Turbo	80286	8	4
Micro Concepts	AT	80286	8	4
Micro Products	XPCAT		8	4
Micro Supply Organization	Super Turbo V20	NEC V20	6	2
MIT Systems	Turbo PC/X		8	6
Mitac	160T	8088-2		2
Multitec Elec.	Multitech 900	8086	8	4
NCR	PC 4	8088	5	2
	PC4i	8088	8	4
	PC 6		6	2
	PC 8		8	4
NEC	APC IV	80286	8	4
	PC-8500			2
Nippon Electronic Tech.	NET V88-2	8088-2	8	4
Olivetti	M18 Desktop	8088	4	2
	M24	80286	6	2
	M28	80286	6	2
OSM	Zeus PC		6	2
	Zeus XPC		5	2
Packard Bell	VT286	80286	8	4
Panasonic	Business Partner	8086-2	6	2
	Executive Partner			2
	Senior Partner		2	1
PC Designs	ET-286i	80286		2
	HD-1000	8088	6	2
	Plain Vanilla	8088		2
PC Engineering	PRO/88 Turbo	8088	8	4
PC Source	Standard 88	8088	8	4
	Turbo 88	8088	8	4
PC Technologies	PCT-AT	8086	8	4
	PCT-AT Turbo	8086	8	4
	PCT-Turbo-XT	80-88	8	4
	PCT-XT	8088	8	4
PC's Limited	286	80286	8	4
	AT	8086	8	4
	Turbo PC	8088	8	4
Philips	3102	8088	4	2
	3200	80286		2
	????	80286	8	4
PMA	PMA PC	8088	8	4
Pro Tech Systems	Turbo PC	8088	8	4
Proteus	286 XTi	80286		2
QIC Research	286-QT	80286	8	4
	Turbo XT	8088	8	4
Sanyo	MBC 675		2	1
	MBC 885	8088	8	4

Seequa	Cobra		6	2
	PC	8088	5	2
	XT	8088	5	2
Siemens	PC 16-05			2
Southern Calif. Systems	SCSI-286	80286	8	4
Sperry	PC	8088	7	3
	PC/HT	8088	5	2
	PC/HT100	8088	5	2
	PC/Micro IT	80286	8	4
Standard	Standard-286	80286		2
	Standard-88	8088	8	4
	Turbo-88	8088	8	4
Stearns	Great Communicator		8	4
Tandon	PCA-20	80286	8	4
	PCI-1	80286	8	4
	PCI-30	80286	8	4
	PCI-40	80286	8	4
	PCX-1	8088		2
	PCX-10	8088		2
	PCX-2	8088		2
	PCX-20	8088		2
Tandy	1000 SX	8088	5	2
	1200 HD PC	8088	5	2
	3000 HD	80286	10	6
	3000 HL	80286	10	6
Tava	Megaplus	8088	8	4
	PC		5	2
Tech PC	PC/XT	NEC V-20		2
	Turbo PC/AT	8086		2
Televideo	2620SH/20		5	2
	AT 1 & 2	80286	8	4
	Tele-PC	8088	1	1
	Tele-XT	8088	1	1
	Telecat 286	80286		2
Texas Indus. Microsystems	IPC-2000	8088	10	6
Texas Instr.	TI Professional			4
THE Computer Prod. Co.	THE PC+	8088	8	4
Toshiba	T-300		7	3
	T3100	80286	5	5
US Data	Industrial PC		10	6
USA Micro	Laser XT	8088-2	8	4
Victor	V286 PC/AT	80286		2
	VPCII PC/XT	8088		2
Vogue Imports	Turbo-Jet PC	8088-2	8	4
	Turbo-Jet PC-2	NEC V-20	8	4
WellsAmerican	A*Star	80286	8	4
WIN Laborat.	ATEGA	80286	8	4
Wyse	WYSEpc 286	80286	8	4
	WYSEpc SY-1100	8088	2	1
	WYSEpc+ WY-1400	8088	2	1
Zenith	Z-148	8088	1	1
	Z-150	8088	4	2
	Z-158	8088	6	2
	Z-248	80286	6	2
	Z-386	80386		4

The "memory map" presented here has been compiled from many sources, including extensive practical experience. It is significant that the region of the PC's memory map that is of most interest to DA&C users is that area with the least available information. "User areas" are open for many potential applications. As such, there is always the possibility of an address conflict. That is, more than one expansion board could be set up to use the same memory locations. The good news is that a very large field of addresses exists, and sufficient space for any practical system can always be found. Reference to this memory map should make address selection easy.

The addresses shown in **boldface** in Figures 3.3D, 3.3E, and 3.3F are suggested memory locations for the installation of PCI-20000 products. However, because each PC configuration can be different, specific addresses can not be predetermined. Given a knowledge of your PC's configuration, this memory map information will allow a large number of suitable locations to be isolated.

Alternative PCI-20000 System Address Locations:

(1) PC/XT only
(2) Without LIM board
(3) PC or AT (not XT)
(4) Usually open
(5) Without EGA
(6) Without EGA
(7) AT, without 128K expansion board

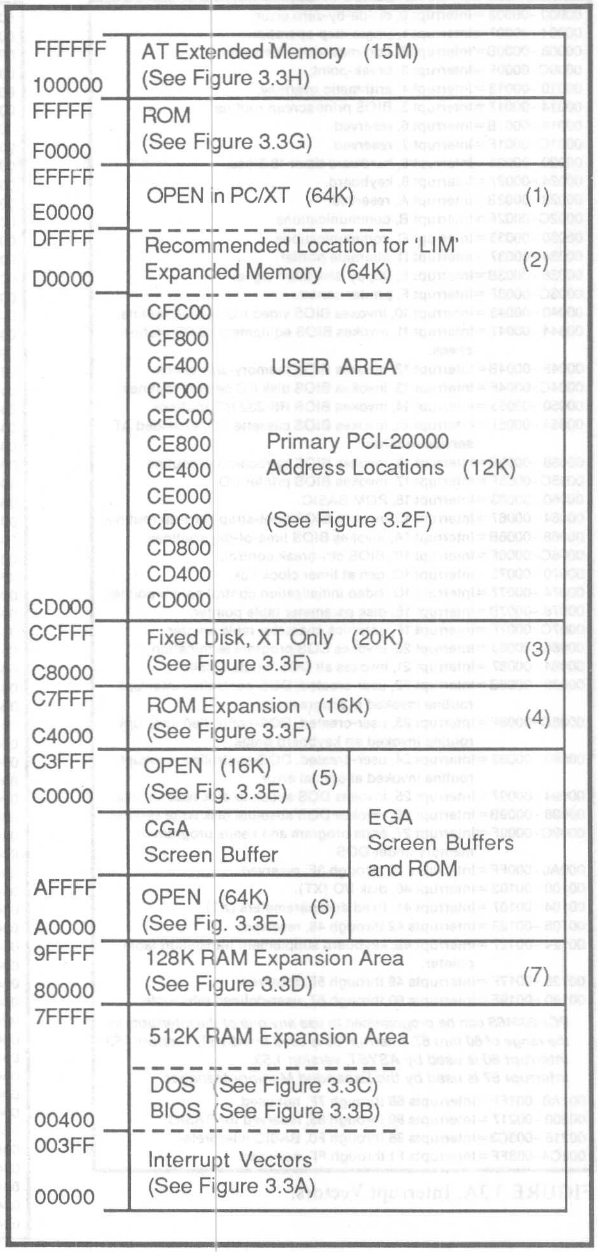


FIGURE 3.3. Memory Map for PC/XT/AT.

00000 -00003	=Interrupt 0, divide-by-zero error.
00004 -00007	=Interrupt 1, single-step operation.
00008 -0000B	=Interrupt 2, non-maskable interrupt.
0000C-0000F	=Interrupt 3, break-point.
00010 -00013	=Interrupt 4, arithmetic overflow.
00014 -00017	=Interrupt 5, BIOS print-screen routine.
00018 -0001B	=Interrupt 6, reserved.
0001C-0001F	=Interrupt 7, reserved.
00020 -00023	=Interrupt 8, hardware timer 18.2/sec.
00024 -00027	=Interrupt 9, keyboard.
00028 -0002B	=Interrupt A, reserved.
0002C-0002F	=Interrupt B, communications.
00030 -00033	=Interrupt C, communications.
00034 -00037	=Interrupt D, alternate printer.
00038 -0003B	=Interrupt E, floppy disk atten signal.
0003C-0003F	=Interrupt F, printer control.
00040 -00043	=Interrupt 10, invokes BIOS video I/O service routine.
00044 -00047	=Interrupt 11, invokes BIOS equipment configuration check.
00048 -0004B	=Interrupt 12, invokes BIOS memory-size check.
0004C-0004F	=Interrupt 13, invokes BIOS disk I/O service routines.
00050 -00053	=Interrupt 14, invokes BIOS RS-232 I/O routines.
00054 -00057	=Interrupt 15, invokes BIOS cassette I/O, extended AT service routines.
00058 -0005B	=Interrupt 16, invokes BIOS keyboard I/O routine.
0005C-0005F	=Interrupt 17, invokes BIOS printer I/O.
00060 -00063	=Interrupt 18, ROM BASIC.
00064 -00067	=Interrupt 19, invokes BIOS boot-strap start-up routine.
00068 -0006B	=Interrupt 1A, invokes BIOS time-of-day routines.
0006C-0006F	=Interrupt 1B, BIOS ctrl-break control.
00070 -00073	=Interrupt 1C, gen at timer clock tick.
00074 -00077	=Interrupt 1D, video initialization control param pointer.
00078 -0007B	=Interrupt 1E, disk parameter table pointer.
0007C-0007F	=Interrupt 1F, graphics character table pointer.
00080 -00083	=Interrupt 20, invokes DOS program termination.
00084 -00087	=Interrupt 21, invokes all DOS function calls.
00088 -0008B	=Interrupt 22, user-created, DOS-controlled interrupt routine invoked at program end.
0008C-0008F	=Interrupt 23, user-created, DOS-controlled interrupt routine invoked on keyboard break.
00090 -00093	=Interrupt 24, user-created, DOS-controlled interrupt routine invoked at critical error.
00094 -00097	=Interrupt 25, invokes DOS absolute disk read service.
00098 -0009B	=Interrupt 26, invokes DOS absolute disk write service.
0009C-0009F	=Interrupt 27, ends program and keeps program in memory under DOS.
000A0-000FF	=Interrupts 28 through 3F, reserved.
00100 -00103	=Interrupt 40, disk I/O (XT).
00104 -00107	=Interrupt 41, fixed disk parameters (XT).
00108 -00123	=Interrupts 42 through 48, reserved.
00124 -00127	=Interrupt 49, keyboard supplement translation table pointer.
00128 -0017F	=Interrupts 49 through 5F, reserved.
00180 -0019F	=Interrupts 60 through 67, user-defined interrupts.

PCI-20046S can be programmed to use any one of the interrupts in the range of 60 thru 67. Interrupt 60 is used by ASYST, version 1.53.
Interrupt 60 is used by ASYST, version 1.53.
Interrupt 67 is used by the Expanded Memory Manager.

001A0-001FF	=Interrupts 68 through 7F, not used.
00200 -00217	=Interrupts 80 through 85, reserved for BASIC.
00218 -003C3	=Interrupts 86 through F0, BASIC interpreter.
003C4-003FF	=Interrupts F1 through FF, not used.

FIGURE 3.3A. Interrupt Vectors.

00400 -00401	=Address of RS-232 adapter 1.
00402 -00403	=Address of RS-232 adapter 2.
00404 -00405	=Address of RS-232 adapter 3.
00406 -00407	=Address of RS-232 adapter 4.
00408 -00409	=Address of printer adapter 1.
0040A-0040B	=Address of printer adapter 2.
0040C-0040D	=Address of printer adapter 3.
0040E -0040F	=Address of printer adapter 4.
00410 -00411	=Equipment flag.
00412	=Manufacturing test indicator.
00413 -00414	=Useable memory size in K.
00415 -00416	=Memory in I/O channel for 64K—planar PC.
00417 -00418	=Keyboard status bits.
00419	=Alternate keyboard numeric input (future use).
0041A-0041B	=Keyboard buffer head pointer.
0041C-0041D	=Keyboard buffer tail pointer.
0041E -0043D	=Keyboard buffer.
0043E	=Floppy disk seek status.
0043F	=Floppy disk motor status.
00440	=Floppy disk motor timeout.
00441	=Floppy disk status.
00442 -00448	=Floppy disk controller status bytes.
00449	=CRT mode code.
0044A-0044B	=CRT column screen width.
0044C-0044D	=CRT regeneration buffer length.
0044E -0044F	=Starting address in regeneration buffer.
00450 -00451	=Cursor position for CRT page 1.
00452 -00453	=Cursor position for CRT page 2.
00454 -00455	=Cursor position for CRT page 3.
00456 -00457	=Cursor position for CRT page 4.
00458 -00459	=Cursor position for CRT page 5.
0045A-0045B	=Cursor position for CRT page 6.
0045C-0045D	=Cursor position for CRT page 7.
0045E -0045F	=Cursor position for CRT page 8.
00460 -00461	=Cursor mode.
00462	=Active page number.
00463 -00464	=Address of current display adapter.
00465	=CRT mode.
00466	=Palette setting.
00467 -00468	=Time count.
00469 -0046A	=CRC register.
0046B	=Last input value.
0046C-0046D	=Low word of timer count.
0046E -0046F	=High word of timer count.
00470	=Timer rollover.
00490 -004CF	=Used by MODE.COM.
00471	=Break indicator.
00472 -00473	=Reboot (Alt-Ctrl-Del) indicator.
00474 -00477	=Fixed disk data area (XT).
00478	=Printer 1 timeout (XT).
00479	=Printer 2 timeout (XT).
0047A	=Printer 3 timeout (XT).
0047B	=Printer 4 timeout (XT).
0047C	=RS-232 card 1 timeout (XT).
0047D	=RS-232 card 2 timeout (XT).
0047E	=RS-232 card 3 timeout (XT).
0047F	=RS-232 card 4 timeout (XT).
00480 -00483	=Additional keyboard buffer pointers (XT).
00484 -004A8	=EGA BIOS buffer.
00484	=Number of character rows.
00485	=Bytes per character.
00487	=Status byte.
00488	=Feature bits, DIP switches.
004A8	=Pointer save.
004D0-004EF	=Reserved.
004F0 -004FF	=intra-application communication area.

FIGURE 3.3B. BIOS Data Area.

00500	=Print-screen status.
00504	=Single-drive status (drive A or B).
00510 -00511	=BASIC's default data segment pointer.
00512 -00513	=IP for BASIC's timer interrupt vector.
00514 -00515	=CS for BASIC's timer interrupt vector.
00516 -00517	=IP for BASIC's ctrl-break interrupt.
00518 -00519	=CS for BASIC's ctrl-break interrupt.
0051A -0051B	=IP for BASIC's fatal-error interrupt.
0051C -0051D	=CS for BASIC's fatal-error interrupt.
00600 -XXXXX	=DOS and "other things".

FIGURE 3.3C. DOS And BASIC Data Area.

7FFFF	= Top of 512K.
80000 -9FFFF	**= AT, 128K RAM expansion area.***
9FFFF	= Top of 640K, end of memory expansion area.

***Suggested memory location for installation of PCI-20000 products.**

FIGURE 3.3D. RAM Expansion Area.

A0000 -AFFFF	**= Enhanced Graphics Adapter (EGA) screen buffers.***
B0000 -B7FFF	= Monochrome adapter or EGA.
B0000 -B0FFF	= Monochrome screen buffer.
B1000 -B7FFF	= Reserved for screen buffers.
B8000 -BFFFF	= Color/graphics adapter (CGA) or EGA.
B8000 -BBFFF	= CGA buffer.
BC000 -BFFFF	= CGA/EGA screen buffers.
C0000 -C3FFF	**=EGA BIOS.***

***Suggested memory location for installation of PCI-20000 products.**

FIGURE 3.3E. CRT Screen Buffers.

C4000 -C7FFF	**= ROM expansion area.***
C8000 -CCFFF	**= Fixed disk control (XT).***
CD000 -CFFFF	**= User PROM, memory-mapped I/O.***
D0000 -DFFFF	**= User PROM, recommended 'LIM' location.***
E0000 -EFFFF	**= ROM expansion area, optional/O for PC/XT.***

***Suggested memory location for installation of PCI-20000 products.**

FIGURE 3.3F. User Area.

F0000 -FDFFF	= ROM BASIC.
FE000 -FFFD9	= BIOS.
FFFF0 -FFFF4	= First code executed after power-on.
FFFF5 -FFFFC	= BIOS release date.
FFFFE -FFFFF	= Machine ID.

FIGURE 3.3G. ROM.

100000 -FFFFFF	= I/O channel memory (PC/AT extended memory, 15Mb maximum).

FIGURE 3.3H. AT Extended Memory.

000 -00F	= DMA controller (8237A).
020 -021	= Interrupt controller (8259A).
040 -043	= Timer (8253).
060 -063	= PPI (8255A).
080 -083	= DMA page register (74LS612).
0A0	= NMI mask register.
200 -20F	= Joystick (game controller).
210 -217	= Expansion unit.
2F8 -2FF	= Serial port (secondary).
300 -31F	= Prototype card.
320 -32F	= Fixed disk.
378 -37F	= Parallel printer (primary).
380 -38F	= SDLC.
3B0 -3BF	= Monochrome adapter/printer.
3D0 -3D7	= Color/graphics adapter.
3F0 -3F7	= Diskette controller.
3F8 -3FF	= Serial port (primary).

FIGURE 3.4. IBM XT I/O Map.

000 -01F	= DMA controller (8237A-5).
020 -03F	= Interrupt controller (8259A).
040 -05F	= Timer (8254).
060 -06F	= Keyboard (8042).
070 -07F	= NMI mask register, real-time clock.
080 -09F	= DMA page register (74LS612).
0A0 -0BF	= Interrupt controller 2 (8259A).
0C0 -0DF	= DMA controller 2 (8237A).
0F0 -0FF	= Math coprocessor.
1F0 -1F8	= Fixed disk.
200 -207	= Joystick (game controller).
258 -25F	= Intel "Above Board".
278 -27F	= Parallel printer (secondary).
300 -31F	= Prototype card.
360 -36F	= Reserved.
378 -37F	= Parallel printer (primary).
380 -38F	= SDLC or bisynchronous communications (secondary).
3A0 -3AF	= Bisynchronous communications (primary).
3B0 -3BF	= Monochrome adapter/printer.
3C0 -3CF	= EGA, reserved.
3D0 -3DF	= Color/graphics adapter.
3F0 -3F7	= Diskette controller.
3F8 -3FF	= Serial port (primary).

FIGURE 3.5. IBM AT I/O Map.

INTERRUPT DRIVEN DATA ACQUISITION WITH THE PCI-20000 SYSTEM

INTRODUCTION

Many computer-based data acquisition applications can benefit from the use of interrupts. Interrupts provide immediate communication from the data acquisition hardware to the computer. They can be used to synchronize data acquisition with external events, to provide prompt response to alarm conditions, and to improve system performance. The PCI-20000 system is designed to make it easy to connect appropriate interrupt signals.

This Application Note is designed to provide an understanding of:

- How microprocessor interrupts work.
- How interrupts are implemented on the IBM PC and compatibles.
- How to determine which data acquisition applications are good interrupt applications.
- How to design software to support interrupts.

Descriptions of sample data acquisition systems using interrupts are included, along with listings of sample interrupt handler routines. Examples show how to program the PC's interrupt controller and how to make use of the IBM PC system clock for data acquisition.

In order to use interrupts effectively, you must follow these steps:

- Analyze your system to determine whether interrupts are, in fact, necessary, and which signals should be used to generate interrupts.
- Connect the selected signals to the computer's interrupt system.
- Write software which will enable the computer to respond to interrupts and to handle the interrupts when they occur.

WHAT ARE INTERRUPTS ?

A microprocessor runs programs by executing **machine instructions** which it reads from memory. Ordinarily, the processor executes instructions sequentially, in the order in which they appear in memory. A special processor register, the **instruction pointer**, keeps track of the next instruction to be executed. Certain instructions, namely **jump** instructions, and certain input signals, namely **interrupts**, cause the processor to start taking instructions from a different area of memory.

A **call** instruction is a special type of jump used to execute subprograms. Before jumping to the new program location, the processor saves the instruction pointer in a block of memory called the **stack**. Another processor register, the **stack pointer**, keeps track of the "top" of the stack. The processor stack is like a stack of plates. Items are added to and removed from the top of the stack. The last item put on the stack will be the first one removed. The call instruction **push**es the instruction pointer onto the stack. This stores the instruction pointer on the top of the stack and updates the stack pointer to show the next location as the top of the stack. The last instruction in a subprogram is a **return** instruction, which **pop**s the instruction pointer off the stack. The stack is restored to its condition prior to the call, and the processor continues executing the instructions following the original call instruction.

An interrupt is a special input signal to a microprocessor. When a transition (usually high-to- low) occurs on the interrupt line, the processor latches the interrupt state and finishes the instruction it is currently executing. If interrupts are **enabled**, the processor then saves the instruction pointer and a word describing its current state on the stack, provides an **interrupt acknowledge** signal, and starts executing a special **interrupt handler** routine. The last instruction of an interrupt handler is an **interrupt return** instruction, which is similar to a return instruction. The original instruction pointer and state of the processor are restored, and the processor resumes executing instructions following the one that was interrupted.

Interrupts can be inhibited during part of a program by executing a **disable interrupt** instruction. If the processor receives an interrupt when interrupts are disabled, it will not respond until it encounters an **enable interrupt** instruction. If an interrupt is pending when an enable interrupt instruction is executed, the processor will then acknowledge the interrupt and execute the interrupt handler routine.

COMMUNICATING WITH EXTERNAL DEVICES

In order for the computer to be useful, the processor must be able to communicate with the outside world. It does this through the keyboard, CRT, disk drives, printer, data acquisition system, and other **peripheral devices**. The processor communicates with the peripherals by reading data from them or writing data to them. Many microprocessors have separate address spaces for **input** and **output (I/O)** and **memory**. A peripheral device can be designed to occupy either I/O addresses or memory addresses. I/O addresses are accessed through **input** and **output** instructions; memory addresses are accessed through **memory load** and **memory store** instructions.

The processor's communication with peripherals is complicated by the fact that the programmer usually can't predict exactly what a peripheral will be doing when the processor reaches a particular place in a program. If the processor attempts to read data from a device when the device hasn't yet supplied the data, the result will be meaningless. If the processor doesn't read the data soon enough, the device may have already supplied new data, or the data may no longer be valid. For example, if two keys were depressed since the last time the keyboard was checked, information about the first key would be replaced by information about the second. Similarly, if the processor attempts to write data to a device that is not ready, the device won't respond as expected.

In order to synchronize communication between the processor and its peripherals, there are two techniques that can be used:

The processor can **poll** a device, periodically reading a status register to determine whether the device requires attention.

The device can be set up to **interrupt** the processor when it needs service.

Both of these methods have advantages and disadvantages which must be weighed for each application.

Polling

The processor polls the device periodically by reading one or more **status registers**, memory or I/O locations whose values allow the processor to determine whether the device needs attention. If the device does needs attention, an appropriate subroutine is called. Otherwise, the processor may continue to poll peripheral devices or it may perform other tasks. A program using polling is usually designed with a single loop containing instructions that poll peripherals and perform all other tasks. A program that handles peripheral devices by polling can be written in any programming language, using ordinary programming skills. Such programs are usually relatively easy to write, understand, and debug.

Although polling is very simple, this method has disadvantages. The processor must always be able to execute the entire loop fast enough to be able to keep up with the demands of the peripherals. A loop which is fast enough most of the time may occasionally fail if too many tasks must be performed on any one pass through the loop. For example, if the processor polls several peripherals, they may occasionally all require service. As more complexity is added to a developing program, a polling loop that originally worked well can become too long. Furthermore, if accurately timed operations must be performed, as is common in data acquisition, a long polling loop may not allow the timing source to be checked often enough to insure adequate accuracy.

Interrupts

With this technique, peripherals signal the processor when they require attention by generating interrupts. Prompt attention to all peripherals is insured as long as demands on the system are reasonable, and the programmer does not need to intersperse polling operations with other program tasks. This method is particularly well suited to an application which requires accurate timing of data acquisition while the processor is performing other operations. Interrupts are also useful if several peripherals requiring service at different rates are used.

In order to make use of interrupts a programmer must write interrupt handler procedures. The addresses of these procedures must be placed in a special location in memory, the **interrupt vector table**, so that they can be executed when an interrupt occurs. This usually requires some knowledge of assembly language. (Some high-level languages provide interrupt handling capability with procedures to perform absolute memory reads and writes, input and output operations, and interrupt routine entry and exit sequences.)

The program flow depends on the occurrence of interrupts and is no longer obvious to the reader of a program listing. Mistakes in handling a computer's interrupt system can result in catastrophic program failures, which makes programs utilizing interrupts especially difficult to debug.

Buffered I/O

A very robust programming technique uses **buffered I/O**, which combines the advantages of polling and interrupts. This technique is particularly useful for applications such as control loops or real-time displays, in which the program must process data as it is being acquired.

An application program using buffered input is written as a polling loop in which the processor waits for data, processes the data, and returns to wait for more data. An interrupt handler routine responds to interrupts to read the data and store it in a **circular buffer**, which is a short array for temporary data storage. The interrupt handler maintains an **insertion counter** to indicate the next position in the buffer which it updates each time data is stored in the buffer. When the counter reaches the end of the array, it is set back to the beginning of the array, completing the circle. A second **removal counter** is maintained by a polling procedure which is called by the application program to read the data buffer. The polling procedure compares the two counters. If they are different, new data exists at the position of the removal counter. The polling procedure updates the counter and returns the new data to the application program. If no new data exists, the polling procedure can be designed to wait for data or to return to the application program with an indication of no data.

If a suitable data acquisition rate is used, the buffer should never contain more than a few entries. However, the interrupt handler must compare the removal counter with the updated insertion counter to detect a possible buffer overflow. If this has happened, the polling procedure should return an error value to the application program.

Buffered output can be handled similarly. The application program passes data to the polling procedure. The data is stored in a circular buffer if there is room for it and rejected otherwise. In this case the polling procedure updates the insertion counter and the interrupt handler updates the removal counter. If data must be updated at regular intervals, as in the case of waveform generation, the buffer must not be allowed to become empty. If the data is being sent to a printer or similar device, the interrupt handler may simply disable the interrupt if the buffer becomes empty.

Choosing the Optimum Strategy

It must be emphasized that using interrupts is not always preferable to polling, and that polling can provide significantly better performance than interrupts for some applications.

The best strategy for high performance (high data acquisition rates) is to use polling and provide a **tight loop** (written in assembly language or machine language) which continuously monitors a status register until the desired condition is satisfied. The peripheral is then serviced, and the processor returns to the tight loop. Unlike the general polling loop described above, this loop performs no other

operations (except possibly to exit the routine after enough cycles have elapsed!).

If precise timing of data acquisition or data output is of primary consideration, and if the timing uncertainty due to a tight polling loop is unacceptable, then timing should be controlled by an external timing source. For example, analog data acquisitions might be triggered by the falling edge of a square-wave pulse train produced by a programmable counter or a frequency generator. The end-of-convert signal from the converter would then be used to signal the processor to read the converted data. Either interrupts or polling could be used, depending on other requirements of the system.

Interrupts are a good way to control data acquisition when the acquisition rate is low enough that there is a significant amount of time available between interrupts. Interrupts can provide a significant advantage over polling if the program must handle more than one device, or if the program must perform other tasks while acquiring data. Most interrupt applications are best handled using buffered I/O.

Examples

It is worthwhile thinking through most applications before deciding to use interrupts, to see if polling could do the job. In many cases there is little or no improvement gained by using interrupts, and polling should be used because of its simplicity and ease of implementation.

An example of a good polling application is a program which must take data very rapidly. A higher sampling rate can be achieved by polling a timing source rather than having the timer interrupt the processor. This is because of the **interrupt overhead**. When an interrupt is detected, the processor must save its status and instruction pointer on the stack and restore them on exit. These operations take much longer than a tight loop which reads and tests a status register until a condition is satisfied. The processor can respond to a timing signal with greater accuracy when a polling loop is used rather than interrupts. This is because the processor must complete the current instruction before acknowledging an interrupt, and the time required for an instruction varies widely.

On the other hand, a program taking data several times per second can make good use of interrupts from the timing source. For such low sampling rates, small variations in the processor's response time to the timing signal are unimportant. Another good application for interrupts is data data acquisition with a slow analog-to-digital converter, using a signal from the converter to interrupt the processor when the conversion is complete. (An integrating converter might require 300mSec for a conversion.) The use of interrupts allows the processor to perform other tasks, such as logging data to disk, performing control functions, or updating a display, while it waits for the peripheral.

Most applications fall somewhere between these extremes. Suppose, for example, that data is to be taken at 20 KHz, or that you are using a converter with an expected conversion time of 50 us. Before deciding whether to use polling or interrupts, you must estimate the interrupt overhead, which depends on the processor and clock speed your computer uses. It may seem wasteful to have the processor spinning its wheels in a polling loop for 50 us, but depending on the interrupt overhead, you may find that it is not possible to improve performance. An example interrupt overhead estimate appears later in this Application Note. The practical limit for sampling rates using interrupts on the IBM PC may range from a few samples per second to a few thousand samples per second, depending on the data acquisition process, other tasks the system must perform, and the programming language used.

8088/8086 FAMILY PROCESSORS AND THE IBM PC

The discussion above is applicable to most modern microprocessors, with minor variations in terminology. We can apply this discussion to the IBM PC and compatible computers with a few qualifications.[1]

The 8086 family microprocessors have 16-bit registers (including instruction pointer and stack pointer registers), but they can address 1 Mbyte of memory, which requires 20 address bits. A complete address is specified by combining the contents of a **segment register** with an **offset register**:

$$\text{Address} = 16 * \textbf{Segment} + \textbf{Offset}$$

The processor has four segment registers, the **Code Segment** (CS), **Stack Segment** (SS), **Data Segment** (DS), and **Extra Segment** (ES). The "instruction pointer" described above is actually formed by combining the IP (instruction pointer offset) register with the CS register. The "stack pointer" is formed by combining the SP (stack pointer offset) register with the SS register.

The IBM PC and compatible computers make use of an **interrupt controller** chip (Intel 8259A[2]) to provide eight different **vectored, prioritized** interrupts.[3,4] The interrupt controller automatically identifies the source of the interrupt and causes the processor to execute the appropriate interrupt handler routine from an interrupt vector table stored in the computer's memory. Because the interrupts are prioritized, high-priority events can interrupt the servicing of low- priority interrupts. Any of the eight interrupts can be inhibited, independent of the processor's interrupt enable and disable capability.

When a high-to-low transition occurs on one of the eight interrupt lines, and that interrupt channel is not inhibited, the controller produces a high-to-low transition on its output line. When the processor responds with the interrupt acknowledge signal, the controller causes the processor to execute a special **software interrupt** instruction. Whenever the controller generates an interrupt, interrupts with lower priority (higher number) are automatically inhibited until the controller is cleared by the processor. The new interrupt condition is latched by the controller, but is not passed to the processor until servicing of the higher priority interrupt is complete. Higher priority interrupts can interrupt a lower priority interrupt handler if the program issues an enable interrupt instruction. The low priority interrupt routine is put on "hold" until the high priority interrupt has been serviced.

The eight hardware interrupt inputs are "mapped" by the controller to software interrupts 8 through 15. A software interrupt instruction appears in assembly language as

INT n

where **n** is a number from 0 to 255. When an **INT n** instruction is executed, the processor saves the program location (IP and CS) and status, and jumps to an interrupt handler whose **address** (offset and segment) is stored at memory location 4*n. For example, **INT 8** would cause the 2-byte word beginning at location 32 (20H) to be loaded into the IP register and the next word (location 34, or 22H) to be loaded into the CS register. The next instruction to be executed will be at this address.

A hardware interrupt can be simulated by putting an appropriate **INT n** instruction in a program. Other software interrupts, which don't correspond to hardware interrupts, are used by the operating system or other programs which are loaded independently but which must be able to communicate with each other. The operating system and its extensions require different amounts of memory depending on the system configuration, so there is no way to predict the exact memory location at which a program will be loaded. It is important to have **absolute** memory locations that can be used for communication between programs. Some of these locations may be used to store information other than the addresses of interrupt handlers. For example, an **interrupt vector** might contain the address of a data table, or it might be used as a "mailbox" to store other information to be passed between programs.

CONNECTING THE PCI-20000 INTERRUPT SIGNAL

The IBM PC interrupt controller has eight interrupt inputs, IRQ0 through IRQ7, which are mapped by the controller to software interrupts **INT 8** through **INT 15**. IRQ0 and IRQ1 are used for system timer and keyboard interrupts. These signals are always generated by circuitry on the computer system board. The other six interrupts are connected to pins in the **I/O Channel,** the slot where expansion boards are plugged in. The IBM PC design reserves most of these interrupts for particular peripherals, but if those devices aren't present, the interrupts can be used for other purposes. PCI-20000 series Carriers permit any of these six interrupts to be selected by placing an appropriate jumper.[5] Of course, no more that one of these jumpers should be in place, and any conflict with other devices should be avoided.

IBM's interrupt signals are assigned as follows:[3,4]

IBM PC I/O Channel Interrupt Signals

Interrupt	IBM Assignment
IRQ2	Reserved
IRQ3	COM2 (second serial I/O port)
IRQ4	COM1 (first serial I/O port)
IRQ5	Fixed disk
IRQ6	Diskette
IRQ7	Printer

The IBM PC-AT has an additional interrupt controller whose interrupts are mapped to INT 71H through INT 77H.[6] The output signal from the second controller goes to IRQ2, which is replaced in the first I/O Channel connector by IRQ9. The operating system calls the code set up for IRQ2 when an interrupt occurs on IRQ9, so that hardware and software designed for the PC will also work on the AT. However, the AT's reset and power-on sequence checks for activity on IRQ9 in order to detect special peripherals. If you use the IRQ2 jumper to select IRQ9 on the AT you must insure that your hardware will never generate interrupts during a reset or power-on, or your computer will be unable to start up. In fact, it's a bad idea to allow any peripherals to generate interrupts during the computer startup.

In addition to selecting the PC interrupt number your system will use, you must install jumpers to connect signals from the I/O Modules to the interrupt line. Before installing these jumpers, be sure that the corresponding module provides appropriate interrupt signals and that your program requires a signal from that module. You may find it helpful to study the examples in the section on Sample Systems. Don't install more module interrupt jumpers than you need, since you will just make it more difficult and time-consuming for your interrupt routine to determine the source of the interrupt.

HANDLING INTERRUPTS IN SOFTWARE

Although your application programs will be mainly written in a high-level language, you will probably have to write at least part of your interrupt handler routines in assembly language or machine language. If you have never programmed in assembly language for the IBM PC, start by writing some practice routines which perform simple tasks, such as adding two numbers and returning the sum to a calling program. This will allow you to become familiar with the 8088/8086 instruction set, the assembler, and the assembly language interface used by your high-level language.

Some programming languages, such as Turbo Pascal[7], make use of **in-line machine language** or **in- line assembly language**. These features allow you to intersperse machine instructions with high- level programming statements. If you will be using in-line code to handle interrupts with a high-level language, you should write some similar practice programs before you begin to program interrupt handler routines.

In order to make use of interrupts, you must provide an interrupt handler to perform whatever tasks you require when an interrupt occurs. DOS provides routines to store and retrieve interrupt handler addresses through the DOS Function Call, INT 21H. For details, you should consult the DOS Technical Reference Manual[8] or the Microsoft MS-DOS Programmer's Reference Manual.[9] If you prefer, you can read and write the interrupt vectors, accessing Segment 0 directly, but you should do this with the processor interrupts disabled.

The interrupt handler must save any registers it uses and set up segment registers that it will require. The interrupt

handler may read and write any memory or I/O location. The interrupt handler should not try to perform any I/O that requires calls to DOS. This normally includes screen output or printer output, keyboard input, and file I/O. All registers must be restored and the interrupt controller cleared before the interrupt return is performed.

After the address of the interrupt handler has been stored, the interrupt controller must be commanded to permit interrupts on the selected channel. At the end of the program, that channel's interrupts must again be inhibited. It's a good idea to make sure the interrupt controller will controller will be re-programmed to its initial state even if the user exits the program by typing CTRL-Break. You can do this by replacing the DOS CTRL-Break function, which is called by INT 23H, by your own "clean up" routine.

The PC operating system does not provide any routines for programming the interrupt controller. The interrupt controller is programmed using the 8088's **IN** and **OUT** instructions. On IBM PCs and most compatibles, the interrupt controller occupies I/O addresses 20H and 21H. The interrupt enable mask is read from or written to I/O location 21H. Bit values of 0 in the mask correspond to interrupt channels that are enabled. The following sequence would enable interrupts on IRQ2 without changing the state of any other interrupts:

```
IN   AL,21H        ;read original mask
MOV  OLD_MASK,AL   ;save mask so it can be restored
AND  AL,0FBH       ;set bit 2 to 0
OUT  21H,AL        ;write new mask
To restore the original controller mask:
MOV  AL,OLD_MASK   ;get original mask
OUT  21H,AL        ;write to controller
```

The interrupt controller must be cleared at the end of the interrupt handler routine. Unless this is done, no further interrupts of equal or lower priority will occur. The following sequence clears the controller:

```
MOV  AL,20H        ;this byte is the end-of-
                   ;  interrupt command
OUT  20H,AL        ;clear the controller
```

USING THE SYSTEM CLOCK FOR DATA ACQUISITION

Many data acquisition applications have relatively low speed requirements and can be adequately timed by the IBM PC's internal system clock. This clock is based on interrupts at approximately 18.2 Hz from an Intel 8253 timer. If you can make use of interrupts at this speed for your application, there are two simple ways to use the internal system clock. With either of these methods, the PC system clock continues to function, so you don't have to worry about maintaining the time-of-day clock, clearing the interrupt controller, or other "housekeeping" tasks performed by the system clock interrupt handler.

Using **INT 1CH**

This is the procedure recommended by IBM for installing a timer interrupt handler. The DOS timer interrupt handler always calls the user timer interrupt handler, **INT 1CH**, before performing an interrupt return. To install your timer interrupt handler routine, retrieve the current interrupt vector for **INT 1CH** and save it. Replace it with the address of your interrupt handler. Your interrupt handler must save all registers used, restore them at the end, and exit with an interrupt return instruction.

Using **INT 8**

This method can provide a more accurate time base than using **INT 1CH**, since your interrupt handler will execute before the DOS routine, which can take varying lengths of time. To replace the DOS **INT 8** interrupt handler with your routine, retrieve the current interrupt vector for **INT 8** and save it. Replace it with the address of your interrupt handler. Your interrupt handler must save all registers used and restore them at the end. Instead of exiting with an interrupt return, however, you will exit with a **FAR JUMP** to the original **INT 8** vector.

Regardless of which of these methods you use, you must be sure to restore the interrupt vector to its original value at the end of the program.

REPROGRAMMING THE SYSTEM CLOCK

If your application requires interrupts faster or slower than 18.2 Hz, you can still use the system timer. However, you must re-program the counter and restore it when your program finishes. The system timer is Counter 0 of an Intel 8253[2] which occupies I/O addresses 40H through 43H on IBM PCs and compatibles. The following sequence will program the counter to interrupt at a new rate:

```
MOV  AL,36H        ;set Mode 3, 16 bits, binary
OUT  43H,AL        ;write control register
MOV  AX,TMR_CNT    ;load new count value
OUT  40H,AL        ;write low-order byte
MOV  AL,AH
OUT  40H,AL        ;write high-order byte
```

The following sequence will restore the counter to its original interrupt rate:

```
MOV  AL,36H        ;set Mode 3, 16 bits, binary
OUT  43H,AL        ;write control register
XOR  AL,AL         ;count value 0 (same as 65536)
OUT  40H,AL        ;write low-order byte
OUT  40H,AL        ;write high-order byte
```

You can determine the value to load the count register with, **TMR_CNT**, as follows:

TMR_CNT = TI * F0

F0 is the frequency in Hz of the input signal to the 8253, which is 1.19318 MHz on the IBM PC and most compatibles. TI is the time interval you want between interrupts. For example, to generate interrupts every 10 mSec,

$$TMR\_CNT = (10 * 10^{-3}) * (1.19318 * 10^6)$$
$$= 11932 \text{ (approximately)}$$

The count you program must be less than or equal to 65536 (0 corresponds to 65536). If you require a larger value, divide that value by a number large enough that the result is less than 65536. Call this number NDIV. Your interrupt handler must then maintain a counter in a memory location which is initially loaded with NDIV and is decremented by 1 on each interrupt. When this counter reaches 0, the interrupt handler should reload it with NDIV and call the data acquisition routine.

For example, to perform data acquisitions every second,

$$TMR\_CNT = 1 * (1.19318 * 10^6)$$
$$= 20 * 59659$$

You would set NDIV to 20 and replace TMR_CNT with 59659.

If you want to maintain the time-of-day clock and the computer's other housekeeping tasks, you must arrange to call the original **INT 8** routine at the right frequency. You can do this by maintaining a counter in a memory location. On each interrupt, the interrupt handler adds the value TMR_CNT to this location. When the result overflows (the addition produces a carry), the system timer interrupt handler should be called. (This procedure results in an average clock frequency the same as that of the original clock, but the rate is not exactly constant.)

The following sequence might form the skeleton of a timer interrupt handler.

```
;These variables must be stored in the code segment:

PC_INT8  LABEL DWORD          ;storage for original interrupt
PC_OFF8  DW    ?              ; offset
PC_SEG8  DW    ?              ; and segment
DSEG     DW    ?              ;storage for data segment value
;--------------------------------------------------------------------------

MY_INT8: PUSH DS              ;save data segment
         PUSH AX              ;save accumulator
                              ;save other registers as needed
         MOV  AX,CS:DSEG      ;load DS with correct value
         MOV  DS,AX           ; (saved in code segment)

;This section of code is required for counts greater than 65536:

         DEC  NDIV_CT         ;decrement NDIV counter
         JNZ  NOT_ZRO
         MOV  AX,NDIV         ;reload NDIV counter
         MOV  NDIV_CT,AX
;--------------------------------------------------------------------------

;This section of code is always required:

         CALL ACQUIRE         ;call data acquisition routine
;--------------------------------------------------------------------------

;This section of code is required whenever the timer has been
;reprogrammed:

NOT_ZRO: MOV  AX,TMR_CNT      ;update dummy system timer
         ADD  SYST_CT,AX      ;ADD sets carry flag on
                              ; overflow
         . . .                ;restore all saved
                              ;  registers
         POP  AX              ;POP doesn't affect flags (in
         POP  DS              ; particular, the carry flag)
         JNC  NOT_CYF         ;skip if timer didn't
                              ;   overflow
         JMP  CS:PC_INT8      ;jump to original routine
                              ;   address (saved in code
                              ;   segment)
NOT_CYF: IRET                 ;return from interrupt
;--------------------------------------------------------------------------
```

3

PERSONAL COMPUTERS IN DATA ACQUISITION
INTERRUPT-DRIVEN SYSTEMS

```
;The above would be replaced by this section of code if the
;timer is not reprogrammed:

        . . .                   ;restore all saved
                                ; registers
        POP  AX
        POP  DS
        JMP  CS:PC_INT8         ;jump to original routine
-----------------------------------------------------------------
```

The symbols **NDIV_CT** and **SYST_CT** refer to memory locations which are allocated in the data segment (not shown). The symbols **NDIV** and **TMR_CNT** may refer to memory locations in the data segment, or they may be constants.

SAMPLE PCI-20000 SYSTEMS USING INTERRUPTS

Following are examples of PCI-20000 systems which might be used for interrupt-controlled data acquisition. Each example includes a brief description of the hardware configuration and the function of the software.

Example 1. Use a PCI-20020M Trigger/Alarm Module to generate an alarm interrupt.

Connect the PCI-20020M *IRQ0 to the interrupt line. When an interrupt occurs, your software will respond to the alarm. The alarm response might include recording the alarm activity, activation of other equipment, or updating a display. The example interrupt handler routine which appears in Listing 1 at the end of this Application Note could be used as a model for the interrupt handler.

This is a good example of an application in which interrupts give better performance than polling. Alarms by definition occur at unpredictable times. Interrupts allow the processor to perform other tasks and still be able to respond quickly to an alarm.

Example 2. Use the rate generator on a PCI-20007M Counter/Timer/ Rate Generator Module to time data acquisition using any combination of I/O Modules.

Connect the PCI-20007M *IRQ0 signal to the interrupt line. Program the Rate Generator to produce an output signal of the desired frequency.[5] When an interrupt occurs, your software will perform the desired data acquisition sequence.

The Turbo Pascal sample program shown in Listing 2 at the end of this Application Note is designed for this system. The program uses a 3000 Hz interrupt signal generated by a PCI-20007M module to time the acquisition of analog data using a PCI-20002M module. While the data acquisition process is occurring the program also graphs the data being acquired. The use of the interrupt handler to acquire the analog data not only allows the program to do two processes at once, but ensures that the acquisition progresses at a well defined rate. Using this method, data acquisition and graphing occur simultaneously without interfering with each other.

Example 3. Use the rate generator on a PCI-20007M Counter/Timer/Rate Generator Module to control data acquisition from a PCI-20019M High Speed DAS Module.

Connect the PCI-20007M SYNC OUT signal to SYNC IN of the PCI-20019M. Connect the PCI-20019M *IRQ0 signal to the interrupt line. You can optionally configure the DAS module for automatic channel advance. When an interrupt occurs, your software will read the converted data from the PCI-20019M. This is a good example of an application in which either polling or interrupts could be used, depending on speed requirements and other program tasks.

TIMING COMPARISONS OF INTERRUPTS AND POLLING

The time required to execute a data acquisition sequence may affect the rate at which data are acquired, and it may determine whether interrupts or polling are to be used to control data acquisition. Following are sample code sequences to support the data acquisition system of Example 3 using interrupts and polling. We will use this example to estimate the time required for a minimum data acquisition process, and to compare the processor overhead required to service an interrupt. The number of 8086 processor cycles required for each instruction is shown in the comment field.[1]

```
;Interrupt entry sequence:
                                ;51 cycles to respond to INT n
MY_INT8: PUSH DS                ;10  save registers
         PUSH ES                ;10
         PUSH DI                ;10
         PUSH AX                ;10
         MOV  AX,DSEG           ;4   load segment registers
         MOV  ES,AX             ;2
         MOV  AX,0C000H         ;4   DS addresses the PCI-20000
         MOV  DS,AX             ;2     carrier segment
         MOV  DI,BUFR_PTR       ;14  load buffer pointer
                                ;127 total cycles for entry
```

```
          ;Data acquisition sequence:
          MOV   AX,CNVT_DAT          ;16  read converter
          STOSW                      ;14  store data in memory and
                                     ;     increment pointer
          CMP   DI,END_BUFR          ;15  check for end of buffer
          JZ    END_PRC              ;4   quit if end of buffer
                                     ;49 cycles data acquisition
          ;Interrupt return sequence:
          MOV   BUFR_PTR,DI          ;15  store buffer pointer
          MOV   AL,20H               ;4   clear interrupt
          OUT   20H,AL               ;10
          POP   AX                   ;8   restore registers
          POP   DI                   ;8
          POP   ES                   ;8
          POP   DS                   ;8
          IRET                       ;32  return
                                     ;93 total cycles for exit
END_PRC: . . .                       ;    quit at end of buffer
```

The total number of cycles of interrupt overhead is 220, corresponding to about 46 uSec for a 4.77 MHz processor clock. The instruction cycle counts shown are for an 8086 processor. These do not give a precise estimate of the actual time required. The 8088 processor used in the IBM PC and compatibles requires more cycles because of its 8-bit bus. Some time is gained by the processor's "pre-fetch queue" which allows it to read the next program instructions during idle bus cycles, and the memory refresh circuitry competes with the processor for bus access. The actual overhead of this example is close to 55 uSec for an IBM PC. (This corrected estimate is based on timing tests of similar code on an IBM PC.)

Compare the interrupt routine above with this polling routine which performs the same function:

```
          ;Setup sequence (performed once):
MY_POLL:  MOV   DI,BEG_BUFR          ;load buffer pointer
          MOV   CX,BUFR_CNT          ;load buffer length
          MOV   DL,MASK              ;get mask for status register
          PUSH  DS                   ;load segment registers
          POP   ES
          MOV   AX,0C000H            ;DS addresses the PCI-20000
          MOV   DS,AX                ;  carrier segment

          ;Data acquisition sequence:
POL_LP:   TEST  STATUS,DL            ;17  test data ready
          JNZ   POL_LP               ;4-16
                                     ;21 cycles loop overhead (min)
                                     ;33 cycles for additional pass
          MOV   AX,CNVT_DAT          ;16 read converter
          STOSW                      ;11 store data in memory
                                     ;     and increment pointer
          LOOP  POL_LP               ;17  continue
                                     ;44 cycles data acquisition
END_PRC: . . .                       ;quit at end of buffer
```

The total loop overhead is 21 cycles minimum, and the data acquisition time is approximately the same as in the interrupt example. Clearly, data could be taken more rapidly using polling than using interrupts. Furthermore, the accuracy with which the processor can respond to a signal is limited only by the polling loop time of 33 cycles. An interrupt response is limited by the cycle time of the interrupted instruction. Typical instructions that access memory require 10-30 cycles, but a multiply or divide instruction might require over 100 cycles. Of course, for maximum timing accuracy, either application would be run with other interrupts masked or disabled. This polling loop is not very versatile: the computer is completely tied up while data is taken. For relatively slow sampling rates, the interrupt overhead becomes negligible, and the interrupt technique allows the processor to perform other tasks.

These sample code sequences contain a few symbols that are not defined above. Instruction timing depends on whether symbols used in **MOV** instructions refer to constants or to memory locations. The symbol **DSEG** is the default data segment value, a constant. The symbol **CNVT_DAT** refers to the memory location of an A/D converter data register. The symbols **BUFR_PTR**, **BEG_BUFR**, **END_BUFR**, and **BUFR_CNT**, refer to memory storage locations in the default data segment.

The examples show two different ways of keeping track of the data count and the location in which the data is to be stored. Assembly language code to set up all the buffer variables might appear as follows, where **BUFR_LEN** is a constant.

```
BUFR_PTR DW    OFFSET BUFFER
BEG_BUFR DW    OFFSET BUFFER
END_BUFR DW    OFFSET BUFFER +
                 BUFR_LEN*2
BUFR_CNT DW    BUFR_LEN
BUFFER DW      BUFR_LEN dup (?)
```

SAMPLE LISTINGS

Listings of two interrupt handling systems are given in this section. These listings, along with the examples and discussions above, can be used as a reference for designing an interrupt handling system to suit your application. The first listing contains several assembly language subroutines which can be adapted for use with any compiled language that produces object files (*.OBJ) that are combined by the linker, LINK.EXE. The second listing is a complete program in Turbo Pascal, in which in-line machine code is used to provide the interrupt-handling functions that are not available in Turbo Pascal.

Listing 1 contains sample interrupt handler routines written in assembly language, which can be assembled by the Macro Assembler and linked with your other program modules by LINK.EXE. The routines must be adapted to interface properly to your compiler. (They were tested with a program compiled by Microsoft C, Version 2.04). Note that other compilers or version numbers may require changes to this code. Please refer to your compiler manual for guidance.

The documentation for your compiler will explain how to interface assembly language modules with compiled programs. You must make sure that the routines treat the registers and stack properly, and that the names of the routines are compatible with your language's naming conventions. The SEGMENT and GROUP declarations[10] must be changed to match the segment and group names used by your compiler. The PROC declarations may need to be changed from NEAR to FAR.

If you use interpreted BASIC, you can not directly use a file generated by an assembler. You must decide how you will load machine-language routines so that BASIC can access them. You might read in the machine instructions from a file or store them in DATA statements in your program. The instructions must be POKEd into memory before the routines can be called. Using machine language subroutines is described in Appendix C of the IBM BASIC manual.[11]

Listing 1 contains the following routines:

SET_INT saves the original values of the interrupt vectors for INT 8 (timer), INT 0AH (IRQ2), and INT 23H (control-break exit function) and sets up a new control-break exit routine.

CLR_INT restores all interrupt routines and re-programs the 8259A interrupt controller and the 8253 timer chips, which may have been altered by the program.

MY_CTBK is the control-break exit routine set up by **SET_INT**. This routine insures that the interrupt routines and peripheral chips will be restored even if the user types control-break to exti the program. It calls **CLR_INT**, then jumps to the original control-break exit routine.

SET_IRQ2 sets up an interrupt handler for IRQ2 and programs the interrupt controller to enable IRQ2.

MY_IRQ2 is the interrupt handler set up by **SET_IRQ2**. It clears the interrupt controller and calls a user-supplied routine to acquire data or perform other functions. **SET_TIM** sets up an interrupt handler for the system timer interrupt, and reprograms the system timer to interrupt at a new rate.

MY_TIM is the interrupt handler set up by **SET_TIM**. It calls a user-supplied routine to acquire data or perform other functions. It maintains the system time-of-day clock by calling the system timer interrupt handler at the proper rate.

Listing 2 shows a complete program, **INTERDMO.PAS**, which includes an interrupt handler routine. This program is written in Turbo Pascal[7], which has provisions for writing inline machine code.

LISTING #1

```
        TITLE   Sample Interrupt Handlers
        .RADIX  16
;*****************************************************************
; IR2_DAT and TIM_DAT
;
; These  are external procedures to acquire data or perform other
; desired functions.  They are called from the interrupt handlers
; for IRQ2 and the timer interrupt (IRQ0), respectively.  Because
; MS-DOS and PC-DOS are not re-entrant,  there are limitations on
; what these routines can do.
;
; IR2_DAT  and TIM_DAT should NOT perform I/O using standard  DOS
; console I/O,  printer I/O, or disk I/O routines.  This includes
; most  I/O  library functions provided with  compiled  languages
; such as C and Pascal.
;
; Many compilers generate code to perform "stack checking" at the
; beginning  of  each subroutine (Microsoft compilers  do  this).
; You  should DISABLE this feature when you compile the  routines
; IR2_DAT  and TIM_DAT,  especially if your program will  perform
; console,  printer, or disk I/O while data acquisition is taking
; place.  Your  compiler  manual should explain how  to  disable
; stack checking.
;
; You should initialize data buffers and other variables used  by
; the IR2_DAT and TIM_DAT before calling SET_IR2 or SET_TIM.
;
```

```
;*********************************************************************
; Language Interface
;
; In order to combine this module with object modules produced by
; a  compiler,  the following SEGMENT and GROUP declarations must
; correspond to the requirements of the compiler.  The segment or
; group  name (depending on the compiler requirements) must  also
; appear  in the OFFSET expressions that appear  in  the  program.
; Some  compilers alter the names of global (public and external)
; symbols.  The names declared PUBLIC and EXTRN in this file must
; match  the names the compiler puts in the OBJ files.   If  your
; linker is case-sensitive, the names must also be in the correct
; case.   You  must refer to your compiler documentation for  the
; correct SEGMENT and GROUP declarations and for the correct form
; of global names.
;
; The  variable ERR_FL,  which must be accessible by the  calling
; program,  is declared external.   The calling program must con-
; tain  a global 2-byte integer variable ERR_FL which is  located
; in the default data segment.
;
; The  external  routines  IR2_DAT and TIM_DAT  are  expected  to
; preserve DS,  SS,  and BP.  The interrupt handlers preserve all
; registers.   All other routines preserve DS,  ES,  SS,  and BP.
; Some languages may require other registers to be preserved.
;
;*********************************************************************
; IBM PC AT
;
; If  you  are  using an IBM PC AT or  equivalent  computer,  you
; should make the following modifications to this module:   Imme-
; diately following EACH "in" or "out" assembly instruction,  add
; the following
;
;          jmp  $+2
;
; This  insures that the I/O chip has adequate time between  suc-
; cessive accesses.
;
pgroup  GROUP   prog
prog    SEGMENT PUBLIC  'PROG'
        ASSUME  cs:prog

INT_0A  DD      ?       ;storage for original interrupt vectors
INT_08  DD      ?
INT_23  DD      ?

D_SEG   DW      ?       ;storage for default data segment
TIMER   DW      0       ;count register
INIT_FL DB      0       ;setup flag
OLD_MSK DB      ?       ;original interrupt controller mask
BUSY_2  DB      0       ;IRQ2 re-entry flag

;*********************************************************************
;
; SET_INT
;
; This routine is intended to be called near the beginning of a
; program. It preserves the original contents of the interrupt
; vectors for IRQ0 (timer, INT 8) and IRQ2 (data acquisition, INT
; 0AH), which will be changed, as well as the original contents
; of the 8259 interrupt enable mask.  A flag, INIT_FL, is checked
; to prevent the setup procedure from being executed twice.
;
; The routines SET_TIM and SET_IRQ2 which set up new interrupt
; handler routines check INIT_FL and call SET_INT if necessary.
;
        PUBLIC  SET_INT
SET_INT PROC    NEAR
        push    bp                      ;save BP
        mov     bp,sp
        sub     sp,2                    ;allocate space for one
                                        ; temporary word
        cmp     cs:INIT_FL,0
        jz      setup
        jmp     set_ex
```


3

PERSONAL COMPUTERS IN DATA ACQUISITION
INTERRUPT-DRIVEN SYSTEMS

```
setup:  mov     [bp-2],ds                       ;save DS in temporary
        inc     cs:INIT_FL
        mov     cs:D_SEG,ds                     ;save DS value in code
                                                ; area
        xor     ax,ax                           ;zero AX
        mov     ds,ax                           ;address Segment 0 with DS

;****** The following is required if the timer interrupt is used.
        mov     ax,ds:(8*4)
        mov     dx,ds:(8*4+2)
        mov     WORD PTR cs:INT_08,ax           ;save original interrupt
        mov     WORD PTR cs:INT_08+2,dx         ; vectors

;****** The following is required if IRQ2 is used.
        mov     ax,ds:(0a*4)
        mov     dx,ds:(0a*4+2)
        mov     WORD PTR cs:INT_0A,ax
        mov     WORD PTR cs:INT_0A+2,dx

        in      al,21                           ;save original interrupt
        mov     cs:OLD_MSK,al                   ; mask

;****** The following is required if either the timer or IRQ2 is
;       used.
        mov     ax,ds:(23*4)
        mov     dx,ds:(23*4+2)
        mov     WORD PTR cs:INT_23,ax
        mov     WORD PTR cs:INT_23+2,dx

;****** Interrupts must be disabled to change interrupt vectors.
        cli                                     ;replace CTRL-BRK with
                                                ; local routine
        mov     WORD PTR ds:(23*4),OFFSET pgroup:MY_CTBK
        mov     ds:(23*4+2),cs                  ;CS contains segment to
        sti                                     ; address MY_CTBK

        mov     ds,[bp-2]                       ;restore DS
set_ex: mov     sp,bp
        pop     bp                              ;retore BP
        ret

SET_INT ENDP

;*********************************************************************
; CLR_INT
;
; This routine must be called prior to the end of the program.
; It restores the original contents of the interrupt vectors
; preserved by SET_INT.  The flag INIT_FL is checked and the
; routine is bypassed if SET_INT was never called.  CLR_INT is
; called from MY_CTBK if the user types a CTRL-BREAK.
;
        PUBLIC  CLR_INT
CLR_INT PROC    NEAR
        push    bp                              ;save BP
        mov     bp,sp
        sub     sp,2                            ;allocate space for one
                                                ; temporary word
        cmp     cs:INIT_FL,0
        jz      clr_ex

        mov     [bp-2],ds                       ;save DS in temporary
        xor     ax,ax                           ;address Segment 0 with
        mov     ds,ax                           ; DS
        cli

;****** Interrupts must be disabled to change interrupt vectors
;       or to reprogram the timer.
```

```
;****** The following is required if the timer interrupt is used.
        mov     al,36                   ;output timer control
        out     43,al                   ; word
        xor     ax,ax                   ;initialize timer count
        out     40,al                   ; to zero
        mov     al,ah
        out     40,al
        mov     ax,WORD PTR cs:INT_08   ;restore original inter-
        mov     dx,WORD PTR cs:INT_08+2 ; rupt vectors
        mov     ds:(8*4),ax
        mov     ds:(8*4+2),dx

;****** The following is required if IRQ2 is used.
        mov     al,cs:OLD_MSK           ;restore interrupt mask
        out     21,al

        mov     ax,WORD PTR cs:INT_0A
        mov     dx,WORD PTR cs:INT_0A+2
        mov     ds:(0a*4),ax
        mov     ds:(0a*4+2),dx

;****** The following is required if either the timer or IRQ2 is
;       used.
        mov     ax,WORD PTR cs:INT_23
        mov     dx,WORD PTR cs:INT_23+2
        mov     ds:(23*4),ax
        mov     ds:(23*4+2),dx

        sti
        mov     ds,[bp-2]               ;restore DS
        mov     cs:INIT_FL,0            ;clear flag
clr_ex: mov     sp,bp
        pop     bp                      ;restore BP
        ret
CLR_INT ENDP

;*********************************************************************
; MY_CTBK
;
; This routine replaces the usual INT 23 "interrupt handler,"
; which is called if a CTRL-BREAK is pressed during execution of
; the program. It ensures that the interrupt routines will be
; restored properly and that the interrupt controller and timer
; chips will be correctly reprogrammed.
;
        PUBLIC  MY_CTBK
MY_CTBK PROC    NEAR
        push    ax                      ;save all registers
        push    bx
        push    cx
        push    dx
        push    si
        push    di
        push    es
        push    ds

        mov     ds,cs:D_SEG             ;load data segment
        call    CLR_INT                 ;restore interrupts

        pop     ds                      ;restore registers
        pop     es
        pop     di
        pop     si
        pop     dx
        pop     cx
        pop     bx
        pop     ax
```

```
        jmp     cs:INT_23                   ;call regular INT 23
MY_CTBK ENDP                                ; routine

;*******************************************************************
; SET_IRQ2
;
; This routine calls SET_INT to preserve the initial interrupt
; routine addresses, if necessary. It sets up a special inter-
; rupt routine for IRQ2 and enables interrupts from IRQ2 by
; reprogramming the interrupt controller.
;
        PUBLIC  SET_IRQ2
SET_IRQ2 PROC   NEAR
        push    bp                          ;save BP
        mov     bp,sp
        sub     sp,2                        ;allocate space for one
                                            ; temporary word
        cmp     cs:INIT_FL,0
        jnz     enable
        call    SET_INT                     ;make sure SET_INT is
                                            ; called
enable: mov     [bp-2],ds                   ;save DS in temporary
        xor     ax,ax                       ;address Segment 0 with
        mov     ds,ax                       ; DS

;****** Interrupts must be disabled to change interrupt vectors.
        cli                                 ;set up IRQ2 routine
        mov     WORD PTR ds:(0a*4),OFFSET pgroup:MY_IRQ2
        mov     ds:(0a*4+2),cs              ;CS contains segment to
                                            ; address MY_IRQ2
        in      al,21                       ;set interrupt mask
        and     al,0fbh
        out     21,al
        sti

        mov     ds,[bp-2]                   ;restore DS
        mov     sp,bp
        pop     bp                          ;restore BP
        ret
SET_IRQ2 ENDP
;*******************************************************************
; MY_IRQ2
;
; This is the interrupt handler for IRQ2. It is written as a
; skeleton which calls the actual data acquisition routine,
; IR2_DAT. (The routine IR2_DAT is not provided in this exam-
; ple.) Since all registers are saved, IR2_DAT may be written in
; a high-level language.
;
; In order to detect a too-rapid data acquisition situation,
; interrupts are re-enabled immediately and the interrupt con-
; troller is cleared. A flag is set to prevent re-entry in case
; one interrupt is not finished before another occurs. If the
; routine is re-entered, it increments an error counter, ERR_FL,
; and immediately returns. The calling program should check the
; error counter as an indication of bad data.
;
        PUBLIC  MY_IRQ2
MY_IRQ2 PROC    FAR
        push    ax                          ;save AX and DS
        push    ds
        mov     ds,cs:D_SEG                 ;set up program data
                                            ; segment
        cmp     cs:BUSY_2,0                 ;test for multiple entry
        mov     al,20                       ;write EOI to 8259 to
        out     20,al                       ; allow interrupt at
        jz      ir2_ok                      ; this level
```

```
        inc     ds:ERR_FL               ;set flag
        jmp     SHORT ir2_ex

ir2_ok: mov     cs:BUSY_2,1             ;set busy flag

;****** Interrupts enabled after testing and setting BUSY_2.
        sti

        push    bx                      ;save all other registers
        push    cx
        push    dx
        push    si
        push    di
        push    bp
        push    es

        mov     es,cs:D_SEG             ;set up extra segment
        mov     bp,sp                   ;save SP in BP

        call    IR2_DAT                 ;call data acquisition
                                        ; routine
        mov     sp,bp                   ;restore SP
        pop     es                      ;restore all registers
        pop     bp
        pop     di
        pop     si
        pop     dx
        pop     cx
        pop     bx

;****** Routine must end with interrupts disabled.  No interrupts
;       can be allowed after BUSY_2 is cleared.
        cli
        mov     cs:BUSY_2,0             :clear busy flag

ir2_ex: pop     ds                      ;restore AX and DS
        pop     ax
        iret
MY_IRQ2 ENDP

;*****************************************************************
; SET_TIM
;
; This routine calls SET_INT to preserve the initial interrupt
; routine addresses, if necessary.  It reprograms the PC's timer
; interrupt to a faster rate (twice the normal rate, as deter-
; mined by TIM_CNT) and sets up a special interrupt routine for
; the timer interrupt.
;
        PUBLIC  SET_TIM
SET_TIM PROC    NEAR
        push    bp                      ;save BP
        mov     bp,sp
        sub     sp,2                    ;allocate space for
                                        ; temporaries
        cmp     cs:INIT_FL,0
        jnz     start
        call    SET_INT                 ;make sure SET_INT is
                                        ; called
start:  mov     [bp-2],ds               ;save DS in temporary
        cli
        mov     al,36                   ;reprogram timer chip
        out     43,al
        mov     ax,TIM_CNT
        out     40,al                   ;output timer count
        mov     al,ah
        out     40,al
```

```
        mov     cs:TIMER,0              ;clear timer register
        xor     ax,ax
        mov     ds,ax                   ;set up local interrupt
                                        ; routine
        mov     WORD PTR ds:(8*4),OFFSET pgroup:MY_TIM
        mov     ds:(8*4+2),cs
        sti

        mov     ds,[bp-2]               ;restore DS

        mov     sp,bp
        pop     bp                      ;restore BP
        ret
SET_TIM ENDP

;*******************************************************************
; Timer interrupt
;
; This is the interrupt handler for IRQ0.  It is written as a
; skeleton which calls the actual data acquisition routine,
; TIM_DAT.  (The routine TIM_DAT is not provided in this exam-
; ple.)  Since all registers are saved, TIM_DAT may be written in
; a high-level language.
;
; In order to maintain an accurate system clock, the PC's timer
; interrupt handler (the original INT 8 routine) is called every
; other time the interrupt occurs.
;
        PUBLIC  MY_TIM
MY_TIM  PROC    FAR
        push    ax                      ;save registers
        push    ds

        push    bx
        push    cx
        push    dx
        push    si
        push    di
        push    bp
        push    es

        mov     ds,cs:D_SEG             ;get program data segment
        mov     es,cs:D_SEG
        mov     bp,sp                   ;save SP in BP
        call    TIM_DAT                 ;call data acquisition
                                        ; routine
        mov     sp,bp                   ;restore SP from BP
        pop     es                      ;restore registers
        pop     bp
        pop     di
        pop     si
        pop     dx
        pop     cx
        pop     bx
        pop     ds

        add     cs:TIMER,TIM_CNT
        jc      tim_ex

        mov     al,20                   ;write EOI to 8259
        out     20,al
        pop     ax                      ;restore AX
        iret                            ;return from interrupt

tim_ex: pop     ax                      ;restore AX
        jmp     cs:INT_08               ;call regular timer
MY_TIM  ENDP                            ; routine

prog    ENDS
        END
```

LISTING 2

```
{ PCI-20000, TURBO-PASCAL, INTERRUPT HANDLING DEMONSTRATION PROGRAM

  HARDWARE: PCI-20001C-2 or -1 (W31 In, W23 In for Interrupt Level 3)
            PCI-20002M-1 (Mod 3), and PCI-20007M-1 (Mod 2) }

{$C-,U-}              { Disable Ctrl-Break                }


Program InterDmo ;


Const

  MaxOff      =   619 ; { Maximum Data Buffer Offset       }
  MinOff      =    20 ; { Minium  Data Buffer Offset       }
  Stat8259    =   $20 ; { 8259 Status Register Port Address }
  Mask8259    =   $21 ; { 8259 Mask   Register Port Address }
  KeysOnMask  =   $FD ; { 8259 Mask for Keyboard ONLY On    }
  EOI         =   $20 ; { 8259 End Of Interrupt Command     }
  ModNo2M     =     3 ; { Module position number of PCI2M   }
  CarSeg      = $C000 ; { segment address of carrier        }
  RGN1        = $0378 ; { N1 for  3000 ticks in 1 second    }
  RGN2        = $0003 ; { N2 for  3000 ticks in 1 second    }
  HdwIntNo    =     3 ; { Hardware Interrupt Number         }

  PgmDS : Integer = 0 ; { Store for Program's Data Segment  }


Type

  DataBuffer = Array [MinOff..MaxOff] of Integer ;  { Buffer for Data }


Var

  Set2M       : Byte Absolute CarSeg:$0302 ;  { Channel and Gain Setup      }
  Strobe2M    : Byte Absolute CarSeg:$0304 ;  { Strobe address to start ADC }
  MSB2M       : Byte Absolute CarSeg:$0304 ;  { Read MSB address for ADC    }
  LSB2M       : Byte Absolute CarSeg:$0305 ;  { Read LSB address for ADC    }
  RGCtrl      : Byte Absolute CarSeg:$0207 ;  { Rate Gen. Control address   }
  RGCnt1      : Byte Absolute CarSeg:$0204 ;  { Rate Gen. Counter 1 addr.   }
  RGCnt2      : Byte Absolute CarSeg:$0205 ;  { Rate Gen. Counter 2 addr.   }
  RGGate      : Byte Absolute CarSeg:$020C ;  { Rate Gen. Enable Gate addr. }

  IntOff      : Integer Absolute $0000:$002C ;  { NOTE :  Must be consistent }
  IntSeg      : Integer Absolute $0000:$002E ;  {         with HdwIntNo.     }

  IntMask     : Byte        ;  { Storage for 8259 interrupt mask             }
  SaveIntOff  : Integer     ;  { Storage for interrupt handler offset address }
  SaveIntSeg  : Integer     ;  { Storage for interrupt handler segment addr.  }
  BufferOff   : Integer     ;  { Pointer into Data Buffer for int. handler    }
  DispOff     : Integer     ;  { Pointer into Data Buffer for grapher         }
  Buffer      : DataBuffer  ;  { The Data Buffer                             }
  Ch          : Char        ;  { Storage for a character pressed on keys      }



{ IntHandler -- this is the interrupt handler. }
Procedure IntHandler ;


Begin  { IntHandler }

  { ***** Save the state of the 8088 CPU ***** }
```

```
InLine ($50/                          { PUSH AX              }
       $53/                           { PUSH BX              }
       $51/                           { PUSH CX              }
       $52/                           { PUSH DX              }
       $56/                           { PUSH SI              }
       $57/                           { PUSH DI              }
       $1E/                           { PUSH DS              }
       $06/                           { PUSH ES              }


       $2E/$8E/$1E/>PgmDS              { CS: MOV DS,PgmDS  ; Setup DS }
       ) ;

 { ***** Process the interrupt ***** }

 If BufferOff <> (MaxOff + 1) Then     { If Buffer is not full }
 Begin

   If BufferOff <> (MinOff - 1) Then   { No read on first pass }
   Begin

     { ***** Get the ADC value ***** }

     Buffer[BufferOff] := (MSB2M shl 4) or (LSB2M shr 4) ;

   End ;

   { ***** Update Buffer Pointer ***** }

   BufferOff := BufferOff + 1 ;

   { ***** Start the next conversion ***** }
   Strobe2M := 0      ;

 End
 Else
 Begin
   RGGate   := 0      ; { Disable Rate Generator }
 End ;
 { ***** Notify 8259 of EOI ***** }
 Port[Stat8259] := EOI ;


   { ***** Restore the state of the 8088 CPU ***** }
   InLine ($07/                        { POP  ES              }
          $1F/                         { POP  DS              }
          $5F/                         { POP  DI              }
          $5E/                         { POP  SI              }
          $5A/                         { POP  DX              }
          $59/                         { POP  CX              }
          $5B/                         { POP  BX              }
          $58/                         { POP  AX              }
          $8B/$E5/                     { MOV  SP,BP           }
          $5D/                         { POP  SP              }
          $CF                          { IRET                 }
          ) ;

 End ;  { IntHandler }
```

```
{ InstallInt -- this routine installs the interrupt handler. }
Procedure InstallInt ;


Begin  { InstallInt }

  { ***** Disable All Interrupts ***** }

  InLine ($FA) ;                    { CLI }

  { ***** Save the current interrupt vector ***** }

SaveIntOff := IntOff ;
SaveIntSeg := IntSeg ;

{ ***** Store the new interrupt vector ***** }

IntOff := Ofs(IntHandler) ;
IntSeg := CSeg            ;

{ ***** Save 8259 mask register ***** }

IntMask := Port[Mask8259] ;

{ ***** Store new 8259 mask ***** }

Port[Mask8259] := KeysOnMask xor (1 shl HdwIntNo) ;

{ ***** Re-Enable Interrupts ***** }

InLine ($FB) ;                    { STI }

End ;  { InstallInt }


{ RemoveInt -- this routine removes the interrupt handler. }
Procedure RemoveInt ;


Begin  { RemoveInt }

  { ***** Disable All Interrupts ***** }

  InLine ($FA) ;                    { CLI }

  { ***** Restore the old interrupt vector ***** }

  IntOff := SaveIntOff ;
  IntSeg := SaveIntSeg ;

  { ***** Restore old 8259 mask ***** }

  Port[Mask8259] := IntMask ;

  { ***** Re-Enable Interrupts ***** }

  InLine ($FB) ;                    { STI }

End ;  { RemoveInt }

Begin  { InterDmo }

  { ***** Save the programs Data Segment Register ***** }

  PgmDS := DSeg ;
```

```
 { ***** Delay long enough for diskette to go off ***** }

 Delay(2000) ;

 { ***** Setup the 2M module to read channel 0 at gain of 1 ***** }

 Set2M := $40 ;

 { ***** Setup the Rate Generator ***** }

 RGCtrl   := $34       ;  { Setup RG Counter 1    }
 RGCtrl   := $74       ;  { Setup RG Counter 2    }

 RGCnt1   := Lo(RGN1) ;  { Write Low  Count LSB }
 RGCnt1   := Hi(RGN1) ;  { Write Low  Count MSB }

 RGCnt2   := Lo(RGN2) ;  { Write High Count LSB }
 RGCnt2   := Hi(RGN2) ;  { Write High Count MSB }

 RGGate   := 0        ;  { Disable Rate Generator }

 { ***** Begin the demonstration ***** }

 HiRes          ;                        { Setting High Resolution Mode }

 InstallInt ;

Repeat

 GotoXY(19,25) ;
 Write(' Press E to Exit, or A to Acquire data ? ') ;

 Repeat

   Repeat                                   { Get A Key }

     Read(KBD, Ch) ;

   Until Not KeyPressed ;

   Ch := UpCase(Ch) ;

 Until (Ch = 'A') or (Ch = 'E') ;

 HiRes          ;                        { Setting High Resolution Mode }

 If Ch = 'A' Then
 Begin

   Draw(  10,    0,  629,    0, 1) ;      { Draw Border               }
   Draw( 629,    0,  629,  186, 1) ;
   Draw( 629,  186,   10,  186, 1) ;
   Draw(  10,  186,   10,    0, 1) ;

   GotoXY(19,25) ;
   Write('      Acquiring and Graphing Data      ') ;

   BufferOff := MinOff - 1 ;
   DispOff   := MinOff     ;

   RGGate   := 3        ;  { Enable Rate Generator }

   Repeat

     If DispOff < (BufferOff - 1) Then
     Begin
```

```
          Draw(DispOff  , Trunc(186-(Buffer[DispOff  ]/4096.0*177)),
               DispOff+1, Trunc(186-(Buffer[DispOff+1]/4096.0*177)), 1) ;

          DispOff := DispOff + 1 ;

        End ;

      Until DispOff = MaxOff ;

    End ;

  Until Ch = 'E' ;

  RemoveInt ;

  { ***** Clean Up the Screen ***** }

  TextMode ;

End.   { InterDmo }
```


References

1. Russell Rector and George Alexy, **The 8086 Book**, Osborne/McGraw-Hill, Berkeley, CA, 1980.

2. **Intel Microsystem Components Handbook**, Intel Corporation, Santa Clara, CA, 1984.

3. **Murray Sargent and Richard L. Shoemaker, The IBM Personal Computer from the Inside Out**, Addison-Wesley, 1984.

4. **The IBM Personal Computer Technical Reference Manual** (6025008), IBM Corporation, Boca Raton, FL, 1981.

5. **The Burr-Brown PCI-20000 Personal Computer Intelligent Instrumentation System User's Manual** (B0C1UMC1.x), Burr-Brown Corporation, Tucson, AZ, 1984.

6. **The IBM Personal Computer AT Technical Reference Manual** (1502494), IBM Corporation, Boca Raton, FL, 1984.

7. **Turbo Pascal, Version 3.0**, Borland International Inc., Scotts Valley, CA, 1985.

8. **Disk Operating System Technical Reference Manual** (1502346), IBM Corporation, Boca Raton, FL, 1983.

9. **Microsoft MS-DOS Operating System Programmer's Reference Manual** (PN 036-014-003), Microsoft Corporation, Bellevue, WA, 1984.

10. **The IBM Personal Computer Macro Assembler Manual** (6172234), IBM Corporation, Boca Raton, FL, 1981.

11. **The IBM Personal Computer BASIC Manual** (6025013), IBM Corporation, Boca Raton, FL, 1981.

DIRECT MEMORY ACCESS (DMA) TECHNIQUES FOR DATA ACQUISITION

INTRODUCTION

Direct memory access and computerized data acquisition are two concepts that have traditionally gone hand-in-hand. There is no faster and more efficient way to get large quantities of data into a personal computer than DMA.

Data acquisition generally involves the monitoring of several sources of physical data at a regular rate defined by a clock or external timing signal. The sources of data are typically mixes of A/D converters, event counters, switches, and contact closures. A/D converters are normally multi-channel devices which convert analog signals from several sources into digital signals for consumption by the computer. Often, some pre-amplification is required to boost the analog signals to the level required by the converter. Event counters, switches and contact closures are already digital signals, and they typically require only buffering to bring their levels to the required value.

Once the mixture of analog and digital input signals has been conditioned into a form acceptable by a computer, they need to be sampled at a regular rate and stored in memory. The three techniques for doing this are polling, interrupts, and direct memory access, or DMA. If the goal is to acquire the maximum amount of data at the highest speed, using the minimum amount of the computer's resources, then DMA is the technique of choice.

The amount of time required to respond to a direct memory access request is infinitesimal compared to the amount of time required to service an interrupt or execute a polling loop. This makes the goals of true background operation and high speed possible. Throughputs of 360kbytes/sec are achievable on an IBM PC or compatible computer using DMA. Burst rates of several megabytes per second are not uncommon among minicomputers. Since DMA is a hardware technique, the only computer resource used is bus bandwidth.

DMA BASICS

The IBM PC's DMA controller contains four separate channels. One channel is used to refresh the machine's dynamic memory, another handles transfers to and from the floppy disk drive, and a third is used to transfer data to and from the hard disk drive, if one exists. This leaves one channel for general use. As with interrupts, the DMA channels are prioritized. The transfers occur so quickly, however, that at rates of less than 100kHz or so, data acquisition would not be impacted.

The DMA controller needs to know where in memory the data from the requesting device is to go (called the 'base address'), and how many items are to be transferred (called the 'byte count'). Then, each time it processes a DMA request, the controller effectively 'steals' a bus cycle from the processor, issues the appropriate address to memory, and sends an acknowledge signal to the requesting device so that it can gate its data onto the computer's data bus. The controller then increments the base address and decrements the byte count for the next request. Since all of this occurs without any software interaction, true background operation is achieved. The computer is free to do any task required, while data acquisition proceeds accurately and invisibly behind the scenes.

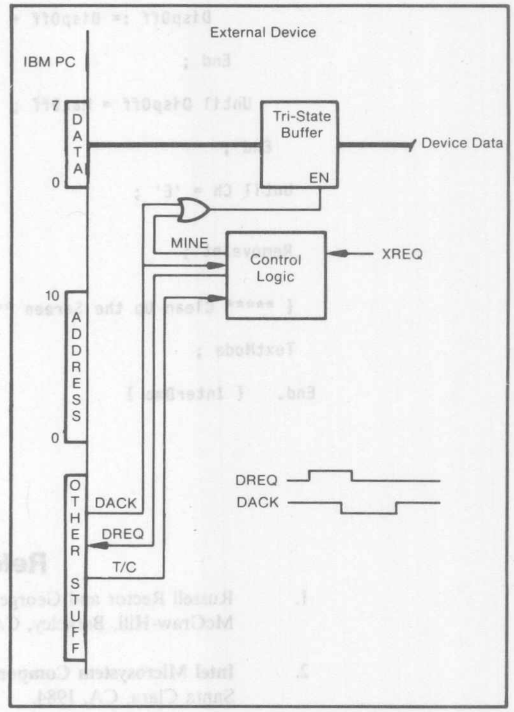


FIGURE 1a. Device-to-Memory DMA Block Diagram.

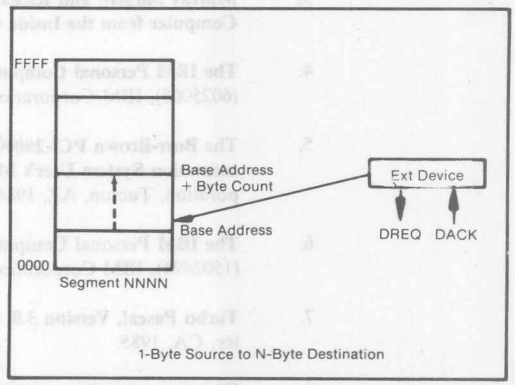


FIGURE 1b. Device-to-Memory DMA Memory Diagram.

LIMITATIONS OF SOME DMA TECHNIQUES

The main drawback to DMA data acquisition is that one can typically only transfer one type of data per DMA channel—usually a sequential group of analog inputs from an A/D converter, so its versatility is limited. Many real applications require a mixture of digital, analog and counter channels. Indeed, most data acquisition systems offer all these data types, but not under DMA control.

The reason for this limitation is that DMA controller chips available today are designed to transfer data efficiently from a single device, or 'pipe' to a large memory buffer in the computer. This is often referred to as 'device-to-memory DMA', and is illustrated in Figures 1a and 1b. Typical applications are tape drive interfaces, disk drive interfaces, local area network interfaces, and high speed communication interfaces. The DMA controller in the IBM PC is of this type.

The DMA controller in the IBM PC/AT can also be used for 'memory-to-memory' DMA. Rather than transfer data from a single source to a block of memory, it can transfer data from one contiguous block of memory to

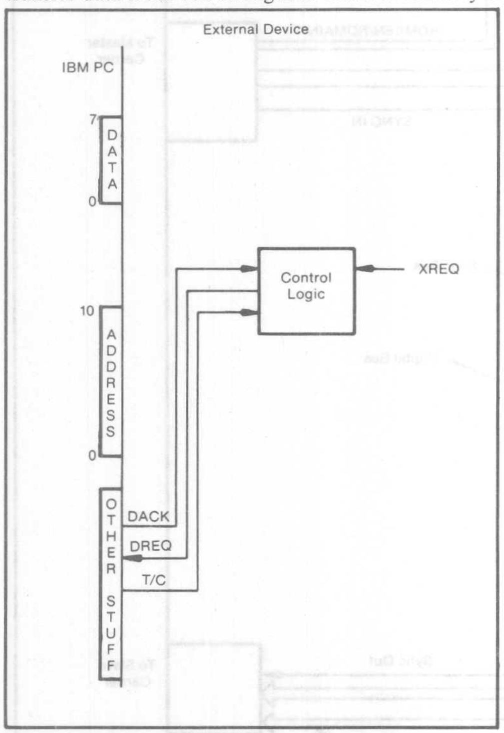


FIGURE 2a. Memory-to-Memory DMA Block Diagram.

another, as illustrated in Figures 2a and 2b. This is useful in graphics controllers, for example, where one may want to transfer a block of memory into a screen buffer.

The device-to-memory DMA technique works fine for most data processing applications. In a data acquisition

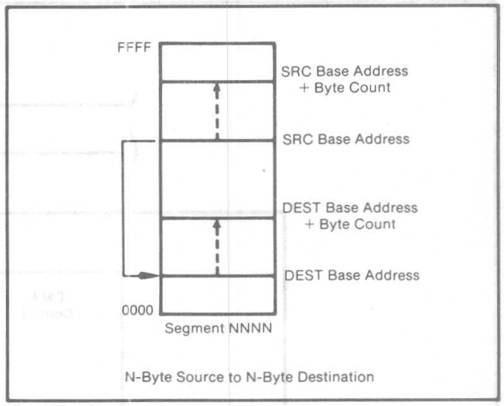


FIGURE 2b. Memory-to-Memory DMA Memory Diagram.

system, however, it limits one to one device per DMA channel. That device must be designed to provide the handshaking required for DMA, in addition to the signals required to operate in in non-DMA mode. The typical approach using the classical device-to-memory DMA technique for data acquisition is to 'hard-wire' a single A/D converter for both DMA and programmed transfer. Usually, the converter has some sort of sequential scanner on its input, allowing multiple channels. The channels to be sampled in the data acquisition run, then, have to be sequential. Some boards provide a 'scan-list' memory for the scanner than will allow the sampling of non-sequential channels.

Memory-to-memory DMA doesn't provide much help for data acquisition. It typically tranfers a large block of memory from one location to another. Both blocks are the same size, and consist of contiguous addresses. For data acquisition, we need to transfer a large group of relatively small 'frames' of random memory or I/O addresses to a large block of contiguous memory. This type of DMA has not existed until now.

A BETTER WAY

A new patent-pending DMA technique has been developed which is targeted directly at efficient DMA transfer for data acquisition systems. The technique has been implemented on the PCI-20041C-3 High Performance Carrier for IBM PCs and compatibles. Using this system, any data type (i.e., digital, analog, counter, etc.) can be put under DMA control simultaneously with any other type. The memory-mapped Carrier occupies a 1kbyte block in the host computer's address space (Figure 3). The board can hold three memory-mapped data-acquisition modules of various descriptions. Each module is allocated 256 bytes of the available 1K. Additionally, the Carrier itself is allocated 256 bytes for its own control functions. All of the functions of both the Carrier and the modules behave as though they were memory locations in the IBM PC. To read the results of an A/D conversion, for example, one would simply read the two memory locations in the Carrier's address space which contain the

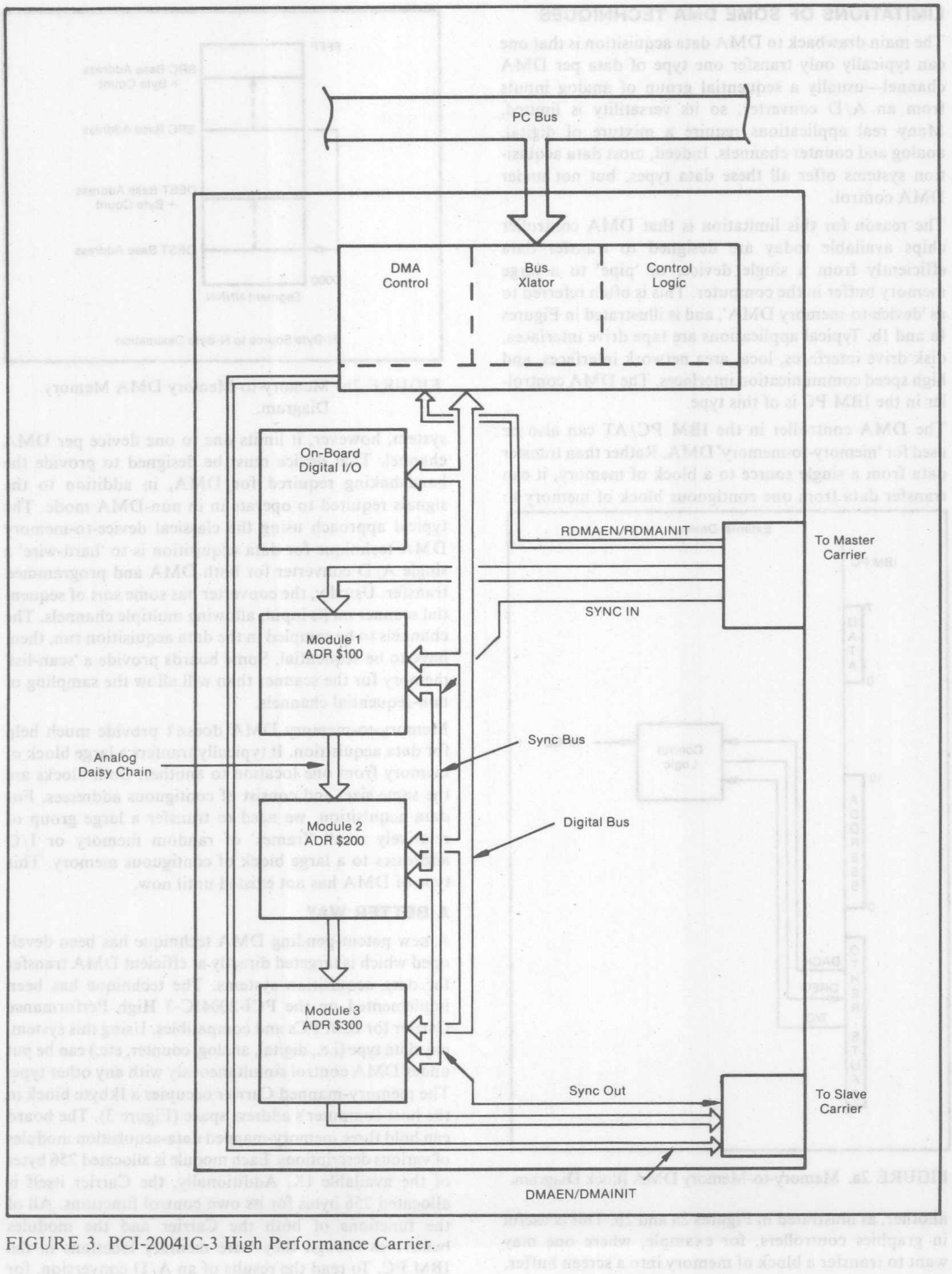


FIGURE 3. PCI-2004IC-3 High Performance Carrier.

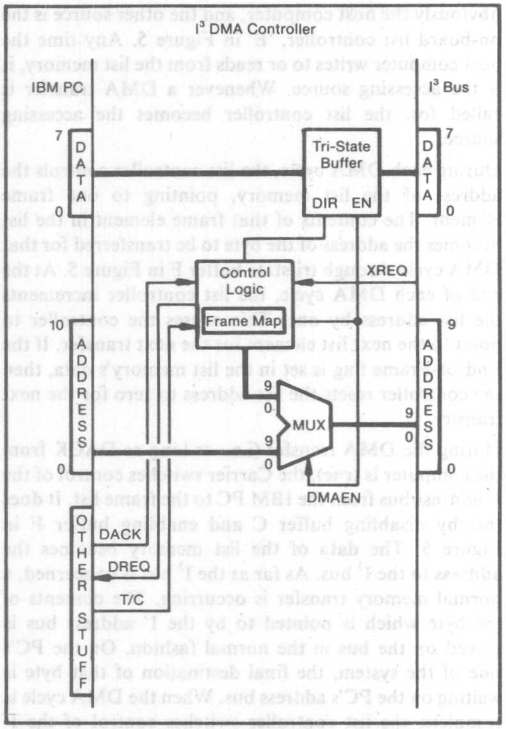


FIGURE 4a. I³ Bus-to-Memory DMA Block Diagram.

two bytes of the conversion. To output a digital I/O byte, merely write the desired value to the address corresponding to the digital I/O byte.

The DMA technique starts with the PC's internal DMA controller, and is illustrated in Figure 4a. Prior to any transfers, it must be programmed to perform a normal DMA sequence. This means programming it for the number of bytes to be transferred, the direction of transfer, the base address of the data to be transferred, and a few other more esoteric things. The IBM PC's controller is a classical device-to-memory controller, so it transfers one byte from the data bus to memory on each DMA cycle. The external device is responsible for insuring that the byte appears on the data bus at exactly the right moment.

The Carrier works with the three DMA signals available on the PC's bus, shown in Figure 5. The first, DREQ, is issued by the Carrier to indicate that it has a byte of data ready to transfer. The second, DACK, is issued by the PC to indicate that the data to be transferred should be put on the PC's data bus (or taken from it, depending on the direction of the transfer). The third signal, T/C, indicates that all the bytes that the controller was programmed for have been transferred. The PC's DMA controller takes care of putting the correct address on its own address bus so that the byte on the data bus falls into the correct memory location. From the PC's vantage point, this looks like classical, straightforward device-to-memory DMA.

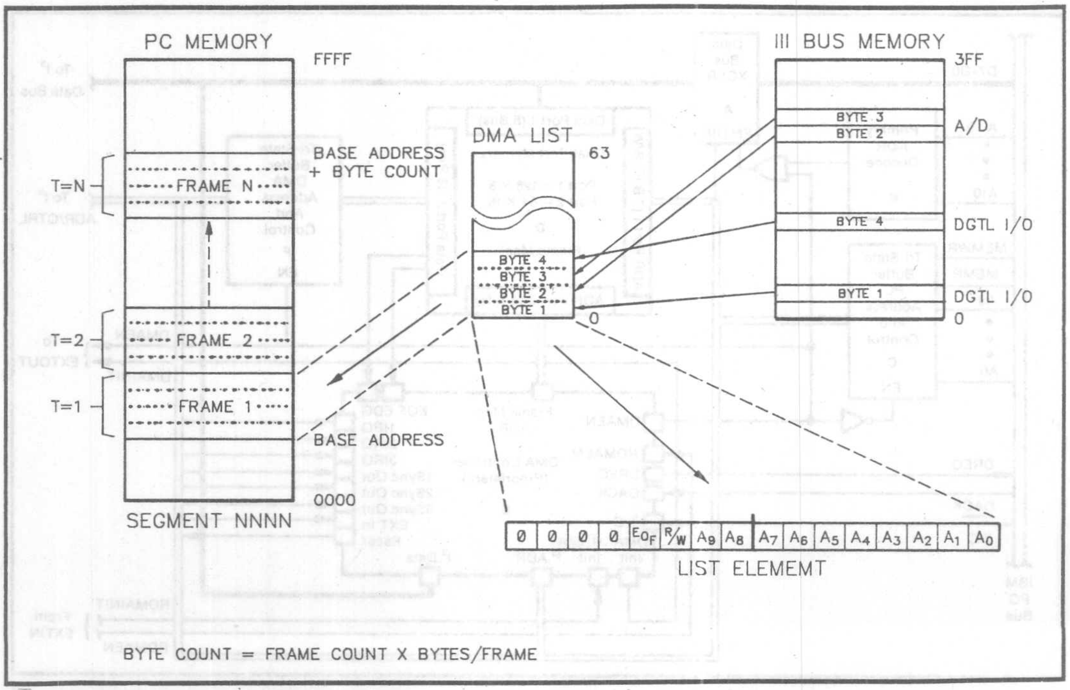


FIGURE 4b. I³ Bus-to-Memory DMA Memory Diagram.

THE BIG DIFFERENCE

128 bytes of the 256 allotted to the PCI-20041C-3 Carrier's functions are set aside as a DMA 'frame map', implemented as a dual-ported memory. The frame map is shown in Figures 4a and 5. This is the key element of the system, and the thing that makes it different. In this map is stored a list of the addresses of all the bytes to be transferred in each DMA frame. For example, if one wanted to acquire data from an analog channel, one byte of digital I/O, and the contents of a 16-bit counter, then one would store the addresses of these items in the map.

In this example, the first two entries in the list would be the addresses of the low and high order bytes of a 12-bit A/D converter module, the third entry would be the address of the digital I/O port, and the last two entries would be the addresses of the low and high bytes of the digital counter. So, the **data** in the list memory is the **address** of the byte to be transferred. Figure 4a shows a logical block diagram of this technique, while Figure 4b shows the resulting memory map.

Each list entry contains the ten-bit address which uniquely identifies a byte in the Carrier's 1Kbyte address space, a Read/Write bit to identify the direction of transfer, and an End of Frame flag to identify the last element of the list. To the host computer, the list looks like a sequential group of memory locations which are read from or written to in the normal fashion. The IBM PC, then, sees the list as 128 8-bit memory locations.

The list memory is dual ported. That is, it can be accessed from two independent sources. One source is obviously the host computer, and the other source is the on-board list controller, 'E' in Figure 5. Any time the host computer writes to or reads from the list memory, it is the accessing source. Whenever a DMA transfer is called for, the list controller becomes the accessing source.

During each DMA cycle, the list controller controls the address of the list memory, pointing to one frame element. The contents of that frame element in the list becomes the address of the byte to be transferred for that DMA cycle through tri-state buffer F in Figure 5. At the end of each DMA cycle, the list controller increments the list address by one. This causes the controller to point to the next list element for the next transfer. If the End-of-Frame flag is set in the list memory's data, then the controller resets the list address to zero for the next transfer.

During the DMA transfer (i.e., as long as DACK from the computer is true), the Carrier switches control of the I^3 address bus from the IBM PC to the frame list. It does this by disabling buffer C and enabling buffer F in Figure 5. The **data** of the list memory becomes the **address** to the I^3 bus. As far as the I^3 bus is concerned, a normal memory transfer is occurring. The contents of the byte which is pointed to by the I^3 address bus is placed on the bus in the normal fashion. On the PC's side of the system, the final destination of that byte is waiting on the PC's address bus. When the DMA cycle is complete, the list controller switches control of the I^3 address bus back to the PC, and increments the frame

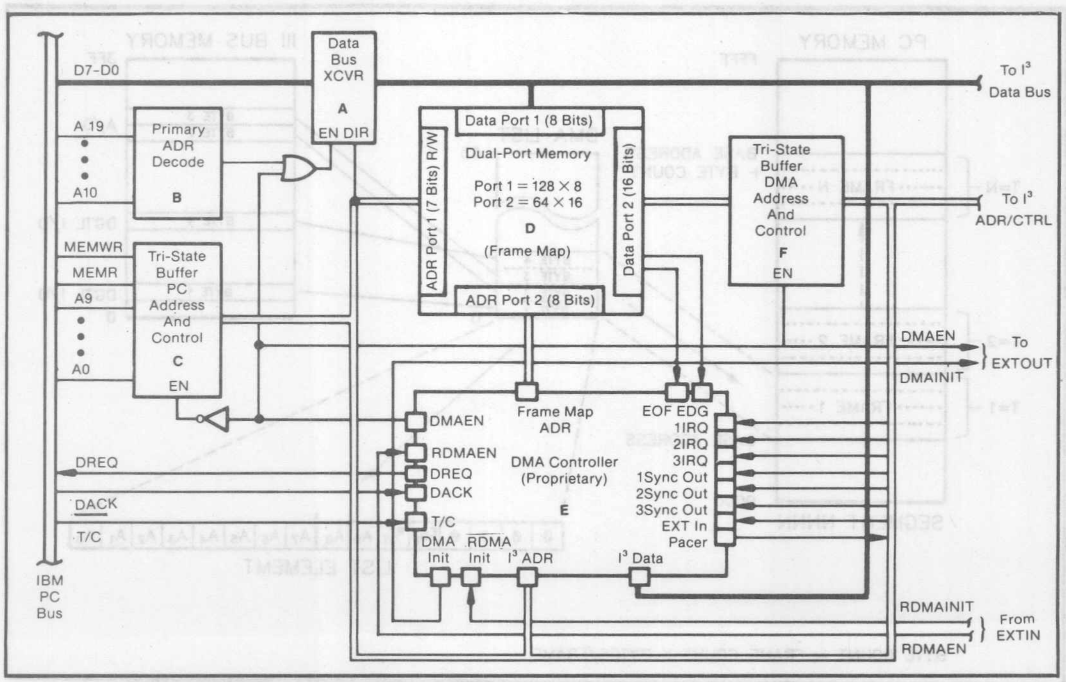


FIGURE 5. DMA System Block Diagram.

list's address counter so that it points to the next element in the list for the next transfer. If the End of Frame flag is set, then the address counter is reset at the end of the transfer so that the list controller is again pointing to the first list element for the next transfer.

The list controller is started by a transaction request signal (XREQ). This signal can come from a variety of places in the system. Each time the XREQ signal is received by the list controller, it will issue one DREQ to the PC and complete the ensuing DMA cycle for each element in the list. So, the transfer of one entire frame of data becomes one indivisible event to the system.

The net result of this scheme is that the list of elements to be transferred by the DMA sequence can be any length up to 64 items, and the addresses of those items can be totally random—they don't have to be sequential at all. The PC believes it is doing normal device-to-memory DMA, and the I^3 bus devices believe they are doing normal programmed transfers. Only the Carrier's DMA controller, E in Figure 5, really knows what is going on.

START/STOP

The system also solves one other problem which is normally encountered in DMA-driven data acquisition. In most data processing applications, the DMA transfer process can be started and stopped by software. If, for example, the computer is transferring one sector buffer from a disk drive to memory, it can totally control the timing of the process. It can start when the computer tells it to, and stop when the data is transferred. In a data acquisition system, this is not always the case.

Data acquisition events tend not be as well-behaved as disk-drive transfers are. They occur asynchronously and the computer has to react to them quickly. Suppose, for example, that it's wanted to monitor a strain gage attached to a steel rod. The steel rod is to be flexed, and one is interested in the strains occurring just before the rod breaks. Obviously, one must acquire data at high speed both before, during, and after breakage. The only problem is that the exact moment of breakage can't be accurately predicted, so it's difficult to know when to start taking data. If one starts too soon, memory fills up before the event of interest occurs. If one waits too long, he misses the 'pre-trigger' data.

This problem is easily solved by specialized start/stop hardware on the Carrier—items J, K, L, and M on Figure 6. The Carrier's control circuitry supports four different methods of starting and stopping DMA. They are:

Mode 1: Start on trigger event after delay, stop on software command.

Mode 2: Start on software command, stop on terminal count.

Mode 3: Start on software command, stop on trigger event after delay.

Mode 4: Start on software command, stop on software command.

Mode 2 is the one used for most data processing applications. The other modes are useful only for data acquisition. Mode 2 is typically the only mode employed in the IBM PC. In data acquisition, it is useful for capturing a block of data at a regular rate from a

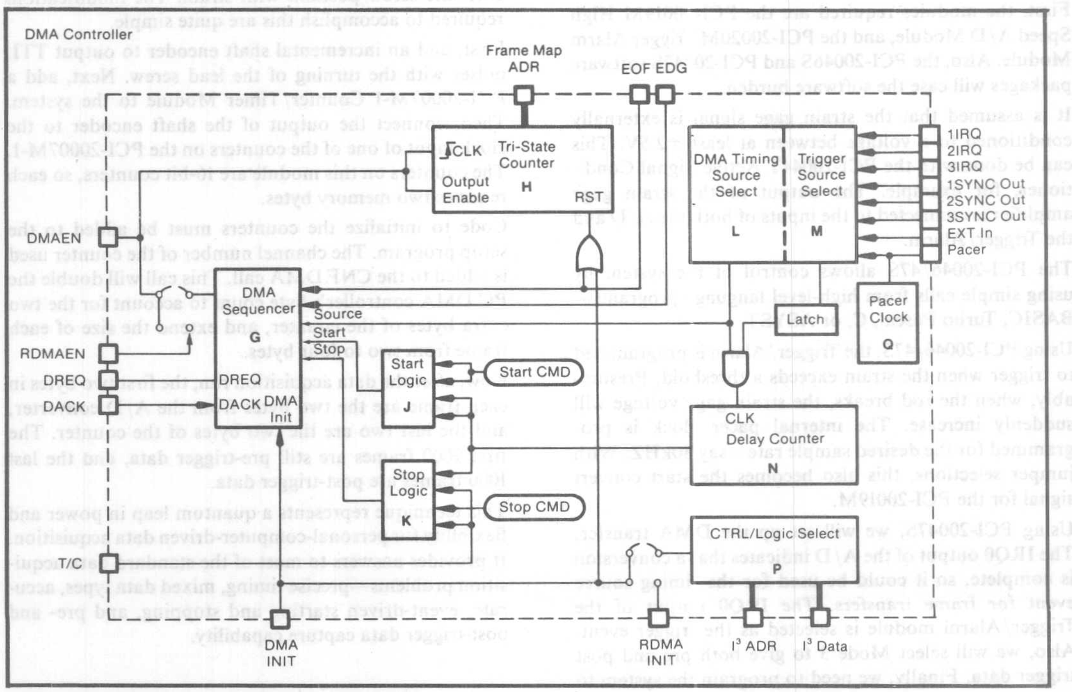


FIGURE 6. DMA Controller Block Diagram.

repetitive signal to be used for digital signal processing, for example. In this case, it is not important for the acquisition to be synchronized to anything.

Modes 1 and 4 are most useful for DMA output. They both involve the use of a circular buffer. In both cases, when the DMA controller has transferred all the bytes in the DMA buffer, it resets its pointers and starts over again. Using either Mode 1 or Mode 4, then, one could build an analog waveform in memory, and then continuously output it through a D/A converter module to develop an arbitrary waveform.

Mode 3 is the most useful for most data acquisition applications, and is the one best used to solve the problem above. Using a circular buffer, Mode 3 provides both pre-trigger and post-trigger information—just right for this application.

There are two significant types of events implied by the four DMA modes above. The first is the trigger event. This is the event which starts or stops the whole series of DMA transactions which comprise a data acquisition "run." The second is the timing source event. This is the source of the XREQ signal which causes each individual frame to be transmitted.

Both of these two signals can come independently from any one of eight sources in the system under software control. The sources are:

- The Syncout signal of any of the three Modules.
- The IRQ0 signal of any of the three Modules.
- The on-board pacer clock.
- The External Syncout from another Carrier.

Here's how to solve the problem of the breaking bar:

First, the modules required are the PCI-20019M High Speed A/D Module, and the PCI-20020M Trigger Alarm Module. Also, the PCI-20046S and PCI-20047S software packages will ease the software burden.

It is assumed that the strain gage signal is externally conditioned to a voltage between at least ±2.5V. This can be done with the PCI-20044T Active Signal Conditioner, for example. The output of the strain gage amplifier is connected to the inputs of both the A/D and the Trigger/Alarm.

The PCI-20046/47S allows control of the system by using simple calls from high-level language programs—BASIC, Turbo Pascal, C, or ASYST.

Using PCI-20046/47S, the Trigger/Alarm is programmed to trigger when the strain exceeds a threshold. Presumably, when the rod breaks, the strain gage voltage will suddenly increase. The internal pacer clock is programmed for the desired sample rate—say 80kHZ. With jumper selections, this also becomes the start convert signal for the PCI-20019M.

Using PCI-20047S, we will set up the DMA transfer. The IRQ0 output of the A/D indicates that a conversion is complete, so it could be used for the timing source event for frame transfers. The IRQ0 output of the Trigger/Alarm module is selected as the trigger event. Also, we will select Mode 3 to give both pre and post trigger data. Finally, we need to program the system to monitor the appropriate A/D channel. All of these selections are made through the "Configure DMA" (CNF.DMA) call of PCI-20047S. It handles setting up all of the hardware on the Carrier, programming the frame map, and programming the PC's DMA controller.

DMA.RUN call will set the number of frames to be acquired, and the trigger delay in frames. This call handles programming the PC's DMA controller for the proper operation, and actually begins the DMA transfer. For our example, we will acquire 10,000 frames, and set the delay to 1000 frames. Thus, our total buffer will consist of 10,000 A/D readings, and the last 1000 will be post-trigger data. After executing this call, data is being acquired by DMA at 80K samples per second.

The machine which bends the rod can now be started, and the data acquisition occurs automatically, filling up the PC's buffer in a circular fashion. When the rod finally breaks, the Trigger/Alarm module will issue the trigger event signal. The delay counter will then count 1000 more A/D samples, and stop the DMA process. The data in the buffer now represents 9000 samples of pre-trigger data, and 1000 samples of post-trigger data.

The data is stored in a special buffer in the PC's memory which the PCI-20047S allocates. It can be retrieved from the DMA buffer and put into a program variable using the READ.DMA call. The first data point in the buffer is the oldest data. In our example, the first 9000 data points are pre-trigger data, and the last 1000 data points are post-trigger information.

Suppose one would like to put a turns counter on the lead screw of the bending machine so that he could correlate screw position with strain. The modifications required to accomplish this are quite simple.

First, add an incremental shaft encoder to output TTL pulses with the turning of the lead screw. Next, add a PCI-20007M-1 Counter/Timer Module to the system. Then, connect the output of the shaft encoder to the clock input of one of the counters on the PCI-20007M-1. The counters on this module are 16-bit counters, so each requires two memory bytes.

Code to initialize the counters must be added to the setup program. The channel number of the counter used is added to the CNF.DMA call. This call will double the PC DMA controller's byte count to account for the two extra bytes of the counter, and extend the size of each frame from two to four bytes.

Now, after the data acquisition run, the first two bytes in each frame are the two bytes from the A/D converter, and the last two are the two bytes of the counter. The first 9000 frames are still pre-trigger data, and the last 1000 frames are post-trigger data.

This technique represents a quantum leap in power and flexibility for personal-computer-driven data acquisition. It provides answers to most of the standard data acquisition problems—precise timing, mixed data types, accurate, event-driven starting and stopping, and pre- and post-trigger data capture capability.

Section 4
Data Conversion Principles

As discussed earlier, digital computers, powerful as they are, speak a very limited "language." Most real-world signals are not in a format (for example: amplitude, level, timing) that can be directly accepted by the computer. It is the data acquisition system that performs the translation function. Internal to the data acquisition unit, there are a variety of data acquisition components that facilitate the translation operations. These include: analog-to-digital (A/D) and digital-to-analog (D/A) converters, multiplexers, sample/holds, amplifiers, counter/timers and some more specialized functions.

Perhaps the most important feature of a data acquisition product is that it brings together these sophisticated functions in a compatible, integrated system. Given the companion software that is available, the user can take advantage of the latest technology without being intimately familiar with the internal details. When selecting a system, however, it is useful to have a basic understanding of data acquisition principles.

Analog Input Systems

The fundamental function of an analog input system is to convert the analog signals into a corresponding digital format. It is the "Analog-to-Digital Converter" (A/D) that transforms the original analog information into computer-readable data (digital, binary code). In addition to the A/D, several other components may be required to obtain optimum performance. These can include: an amplifier, a sample/hold, a multiplexer and signal conditioning elements.

Analog-to-Digital Converters - A significant number of different types of A/D converters exist today. Among these, a few stand out as the most widely used: successive approximation, integrating and parallel (flash) converters. While flash converters are the fastest, they are also the most expensive. Complexity generally limits these devices to low-resolution (8 bits or less) applications. Most data acquisition tasks usually require a minimum of 12-bit resolution. In fact an increasing number of tasks may need 14- to 16-bit resolution. It is predictable that higher resolution converters are not only more expensive but they are usually slower. Therefore, it makes sense to carefully consider the requirements before making a "resolution" decision.

A good starting point is the input sensor or transducer. Some sensors have very wide dynamic ranges. Dynamic range is the span, or difference, between the maximum full scale signal level and the lowest detectable signal. There is not necessarily a good correlation between sensor accuracy and dynamic range. For example, a 0.5% accurate transducer can have a dynamic range of more than 80 dB. This requires a system with at least 12-bit resolution. To maintain maximum dynamic range, some applications may require 14- to 16-bit resolution. It is interesting that an application with 16-bit range does not necessarily require a 16-bit A/D converter. A programmable gain amplifier can yield increased resolution. Amplifying a low-level signal by 10 or 100 increases the effective resolution by more than 3 and 6 bits respectively. Starting with a 12-bit converter this results in 15 to 18 bits of dynamic range. A related technique is described in Section 9, under "Getting Increased Resolution From A 12-bit A/D Converter."

A 12-bit system provides a resolution of one part in 4096 (2^{12}) or approximately 0.025% of full scale. 16 bits corresponds to one part in 65536 (2^{16}) or approximately 0.0015% of full scale. Therefore, resolution not only determines dynamic range but it also limits overall system accuracy. On the other hand, increasing a system's resolution cannot benefit its accuracy if other components such as the amplifier or sample/hold are the limiting factor.

When an input signal change is smaller than the system's minimum resolution, then that "event" will go undetected. For instance, when using a 12-bit A/D converter (without any pre-amplification), any signal change that does not exceed 2.44 millivolts on the 10 volt range will not be "seen" by the data acquisition system. In contrast, if the signal is first amplified by 1000 before conversion, the resolution could be increased to 2.44 **microvolts** (in the absence of noise).

For speeds above 100 samples/second the successive approximation converter is most popular. In fact, speeds above 100K samples/second are attainable. Binary weighted "guesses" are compared to the actual input signal until a match is achieved. It is essential that the input signal remain constant during the course of the successive comparisons or very significant errors can result. This requires the use of a sample/hold circuit, as described below.

When high speed is not required, an "integrating" A/D converter can give 12-, 14- or even 16-bit resolution at low cost. Sampling speed is typically on the order of 3 to 50 conversions per second. As the name implies, this converter averages any input signal variations during the conversion cycle. This feature inherently filters input noise. Linearity and overall accuracy are generally better than in the other A/D converters.

Accuracy is an important measure of an analog input system. It defines the total error in any particular reading. For example, a data acquisition system specified as accurate to 0.05% of full scale on the 10 volt range, would exhibit a worst-case error of 5 millivolts ($10V \cdot 0.0005$). If the system is specified as 0.1% accurate on the ±10 millivolt range (for example the A/D on the $\pm10V$ range and the PGA in a gain of 1K) the system would exhibit a worst-case error of 20 microvolts referred to the input ($20mV \cdot 0.001$). In assessing the value of a data acquisition system, the accuracy specification requires careful scrutiny. Be sure the accuracy is specified for the input range of interest.

Amplifiers - Analog input signals can vary in amplitude over a very wide range. The A/D converter, however, requires a "high level" signal in order to perform at its best. In many systems an amplifier is provided to boost possible "low level" signals to the desired amplitude. Ideally, the input amplifier will have several gain choices available, all under software control. This device is usually called a programmable gain amplifier (PGA). However, cost and performance trade-offs sometimes dictate that the gain of

the amplifier should be "manually" adjusted. Manual adjustment refers to the selection of a resistor or a jumper.

A simple analog input stage is shown in Figure 4.1 Remember, the amplifier shown in this diagram may not be required in every application. As shown, this circuit can accommodate only one input channel. One way of measuring several channels would be to duplicate the A/D converter and amplifier for each input signal. However, there is a less expensive way described below.

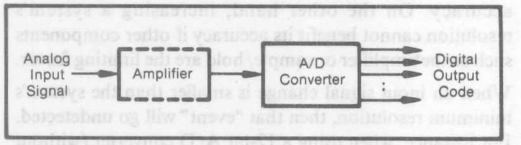


FIGURE 4.1. An Analog Input Channel.

Multiplexers - The Multiplexer (Mux) shown in Figure 4.2 is simply a switch arrangement that allows many input channels to be serviced by one amplifier and A/D. Software can control these switches to select any one channel for processing at a given time. This approach offers considerable cost savings over separate amplifiers and A/D's. Since the amplifier and A/D are being shared, the speed of analog acquisition will be reduced. To a first approximation, the rated "speed" of the amplifier and A/D will be divided by the number of input channels serviced. "Throughput" is often defined as the "single-channel speed" multiplied by the number of channels.

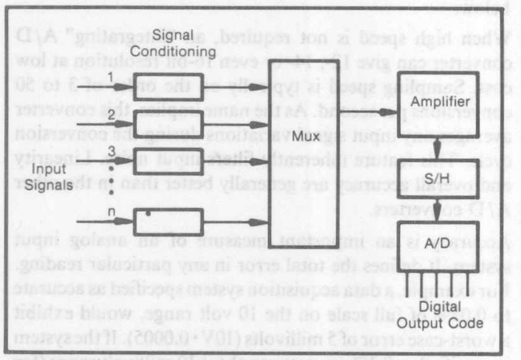


FIGURE 4.2. A Complete Analog Input Subsystem.

Sample/Hold - In general, an analog input signal can be changing with time. That is, the input could be an AC signal whose amplitude varies continuously. Successive approximation A/D converters require that the input amplitude not change during the conversion cycle. The function of the Sample/Hold (S/H) is to "grab" the present value of the signal just before the beginning of an A/D conversion. This level is held constant, despite a changing input, until the A/D conversion is complete. This feature allows the accurate conversion of high-frequency signals.

Time Multiplexing - The system described in Figure 4.2 shares the amplifier, S/H and A/D converter between the various input channels. The user selects the desired sample rate to fit the given application. If each channel is to be read "R" times per second, then the Mux must scan at "n" times this rate (where, "n" is the number of channels to be read). Clearly, the S/H and A/D must be fast enough to allow a complete conversion in less than $1/(R \cdot n)$ seconds.

We must be careful not to be misled by the speed specifications of the individual components in the system. "Conversion time" defines only the speed of the A/D converter, which is only part of the total time required to measure a given channel. In order to understand the true speed of a system we must know either the "per channel sample rate" or "throughput rate," and the conditions under which it was specified (for example, throughput is a strong function of the amplifier gain).

Ideally, all of the input channels will be read at the same time, every 1/R seconds. However, time multiplexing inherently generates a "skew" or time difference between each channel's reading. If the Mux, S/H and A/D combination are "fast enough," then it may appear that the channels are being read at the "same" instant. Some applications are very sensitive to time skew, such as the measurement of instantaneous electrical power ($I \cdot V$), or relative position of mechanical components. Given the fastest A/D converters available, there are still many applications that cannot tolerate the time difference between readings resulting from sequential readings. In critical applications the technique of "Simultaneous Sample & Hold" can further reduce time skew by a factor of 100 to 1000 times.

The simultaneous sample/hold architecture is ideal for applications in which the phase and time relationships of multiple input channels are critical to the given investigation. For example, if the system in Figure 4.2 were sequentially scanning four analog inputs at a throughput of 89K samples/second, the time elapsing between conversions would be 11.25 microseconds. About 45 microseconds will be required to digitize all four channels. This represents a 162 degree phase shift between the first and fourth channel at a 10KHz signal frequency (45us/100us $\cdot$ 360 deg). In contrast, the simultaneous sample/hold system in Figure 4.3 can capture all four channels within about 10 nanoseconds of each other, representing a phase shift of less than 0.04 degrees at 10KHz.

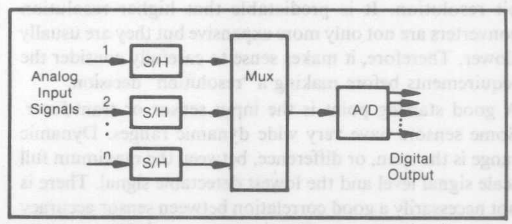


FIGURE 4.3. Simultaneous Sample & Hold Systems.

In addition to phase and time relationships, this technique is particularly useful for applications in which cross-correlation functions must be calculated. Prime examples include: speech research, materials and structural dynamics testing, three-phase electrical power measurements, geophysical signal analysis, and automatic test equipment (ATE) on production lines.

Signal Conditioning - Even with the high quality components mentioned above, it may be desirable to preprocess the input signals. This task is called signal conditioning, and is often divided into two categories. "Active" signal conditioning can include amplification and isolation, while "passive" signal conditioning includes voltage division, surge suppression, current-to-voltage conversion and filtering.

The maximum signal amplitude that can be applied to an amplifier or A/D convertor is usually ±10 volts. Resistive dividers can be used to scale virtually any voltage level down to this acceptable range. Thus, monitoring 48 volts or 480 volts is entirely practical. It is important to consider that the multiplexer and other electronic components can be permanently damaged if signals above 15 volts make direct connection to these devices. The addition of clamping devices such as zeners or MOVs to the signal conditioning network can insure protection against possible input faults or surges.

It is sometimes desirable to preamplify low-level signals (1mV to 1V) outside the main DA&C enclosure to maintain maximum signal-to-noise ratio. One form of this kind of signal conditioning is the "two-wire transmitter." Transmitters not only amplify the input signal but can also provide isolation, linearization, cold-junction compensation and conversion to a high-level current (typically 4 to 20 milliamps). Current transmission allows signals to be sent up to several thousand feet (1500 meters) without significant loss of accuracy. While voltage signals are rapidly attenuated by the resistance of the connecting wires, current signals are not. In a current loop, the voltage drop due to wire resistance is compensated for by the compliance of the current source. That is, the voltage across the current source automatically adjusts to maintain the desired current level.

When signals are represented by currents, it is a simple matter to convert them to voltage signals with a resistor. Values of 250 to 500 ohms are most common (producing 5 to 10 volt signals, for 4 to 20mA currents).

Of all the signal conditioning categories, filtering is the most widely needed, most widely used and most widely misunderstood. Simply stated, filtering is used to separate desired signals from undesired signals. Undesired signals include: noise, AC line frequency pick-up, radio/TV station interference, and signal frequencies above 1/2 the sampling frequency. Generally, a low-pass filter is employed to control these unwanted sources of error, by excluding the portion of the frequency spectrum where desired signals do not exist. When input signal frequency components above 1/2 the sampling frequency are allowed at the input to the A/D converter, a phenomenon known as "aliasing" occurs. This results in the generation of spurious signals within the frequency range of interest that can not be distinguished from real information. Hence, serious errors in the interpretation of the data can occur. This discussion of signal conditioning is intended only to suggest the need for this type of consideration. More detail will be offered in Section 8 on "Signal Conditioning."

Single-Ended vs Differential Signals - Analog signals can be configured as either single-ended or differential inputs. Single-ended inputs all share a common return or ground line. Only the "high" ends of the signals are connected through the multiplexer to the amplifier. The "low" ends of the signals return to the amplifier through the system ground connections. That is, both the signal source and the input to the amplifier are referenced to ground. This arrangement works fine as long as the ground potential difference is very small. Problems arise when there is a large difference in ground potentials. This causes extraneous currents to flow (a "ground loop") which can generate errors. The main advantage of single-ended inputs is the low per-channel cost. Only one multiplexer switch is required to handle each input channel.

A differential arrangement allows both the non-inverting (+) and the inverting (-) inputs of the amplifier to make connections to both ends of the actual signal source. In this way, any ground-loop induced voltage appears as a common-mode signal and is rejected by the differential properties of the amplifier. While differential connections can greatly reduce the effects of ground loops, they require two multiplexer switches per channel. Thus a 32 channel, single-ended system can handle only 16 differential inputs. In addition, while a simple op-amp can be used for single-ended inputs, an instrumentation type amplifier is required for differential inputs.

In some applications the so-called "pseudo-differential" connection can be employed. This is actually a single-ended connection in which one of the inputs is connected to the common ground return point of the input signals. Thus, this channel measures the ground-loop induced voltage which can then be corrected for in the software. This technique is useful when all of the input signals are referenced to the same ground potential.

Instrumentation Amplifiers (IA) - As suggested above, the instrumentation amplifier is a differential input gain block that presents a very high impedance at both the + and - input terminals. The common-mode rejection characteristics attenuate the effects of ground loops, AC power line pick-up and noise induced error signals. Thus, the IA is especially useful for measuring low-level signals. When the IA has software programmable gain, it is known as a PGIA (programmable gain instrumentation amplifier.) Because virtually all programmable gain amplifiers in DA&C systems are IA's, we simply refer to PGIA's as PGA's.

Ideally the input impedance, common-mode rejection and bandwidth of amplifiers would be infinite. In addition, input current and offset voltage would be zero. This implies that the measuring circuit does not influence the signal source. However, real amplifiers do have finite input impedance and input current characteristics as well as offset voltage (Vos). Offset voltage refers to the amplifier's output voltage when zero input is applied (inputs are shorted). Actually, Vos is the input voltage that must be applied to the ± input of the amplifier to make the output voltage zero. Offset voltage is due to small mismatches in the characteristics of components in the amplifier's input stages. While most amplifiers have provisions for trimming the offset to zero, this is not done without sacrificing other parameters. For example, trimming Vos often generates an additional amount of offset drift (Vos change with temperature) and other non-ideal effects. Vos can often be compensated for

in the software. The PCI-20000 has built-in provisions for offset correction.

In most cases it is the input current that is potentially most troublesome. Two terms are used to describe input current: bias current (Ib) and offset current (Ios). Bias current refers to the current flowing into (or out of) either the + or - terminal of the amplifier. Offset current is the difference between the + and - bias currents. In principle, the distinction is important because Ios can be much smaller than Ib. These non-ideal currents interact with the signal source impedance to produce an additional offset voltage term. When the source impedance is balanced, that is, equal at both + and - inputs, it is only Ios that generates an error.

It is essential that an external conductive path exists between the input terminals of the amplifier and its power supply ground. In addition, the resistance of this path must be small enough so that the resulting offset voltage (Ib•Rs) does not interfere with the amplifier's performance. In the extreme case where the inputs are left floating (no external return resistance), the amplifier is likely to reside in a nonlinear or otherwise unusable state. As a general rule, single-ended inputs do not require attention to the bias current return resistance. This is because one side of the input is directly connected to ground and the other input has a return path through the signal source. In contrast, differential connections almost always require the user to provide an external return resistance path. Normally the DA&C system's "termination panels" have provisions for these resistors. Typically, values of 100K or 1 Megohm are used.

Analog Outputs

In many applications analog output signals are required. These signals are used to drive chart recorders, to provide feedback in closed-loop control and to initiate a variety of other tasks. Common analog output ranges include ±5V, ±10V, 0-10V and 4-20mA.

When operating in the voltage output mode, most D/A converters can supply up to 5 10mA of load current. However, some multiple output systems have ratings as low as 1mA. This is not usually a limitation, because the majority of these applications call for driving high impedances. When large loads such as positioners, valves, lamps and motors are to be controlled, power amplifiers or current boosters are required. Most DA&C systems do not include high power analog drivers within the standard configuration.

Digital Inputs and Outputs

Most data acquisition systems are able to accept and generate TTL level signals (0 to 5 volts). However, applications often require an interface for other discrete voltage levels. Higher voltage and current outputs are also required to control devices such as solenoids, motors and relays.

A number of standard signal termination panels are available to facilitate the connection of the field wires to the DA&C system. These termination panels have provisions for screw terminal connections, signal conditioning, channel status indicators (LEDs), voltage dividers and isolators. Thus, the monitoring and control of high DC levels, along with AC line voltage circuits, are readily accomplished.

These features will be described in more detail in the Signal Conditioning Section of this handbook.

Pulse and Frequency Inputs and Outputs - A variety of counting, timing and frequency measuring applications exist. Other applications require that devices be turned on and off for precise time periods. All of these functions can be provided by "counter/timer" (C/T) circuits. The system's counter/timers are optimized for pulse applications including frequency measurement and time-base generation. Counters are characterized by the number of input events that can be accumulated and by their maximum input frequency. Most systems employ 16-bit counters that can accumulate pulses at frequencies up to 8 megahertz. Up to 65536 (2^{16}) events can be accumulated before the counter overflows. The counters are all independent of each other and can be used to count events, measure frequency or act as frequency dividers. The pulse generators (rate generators) are software programmable over a very wide range of frequencies and duty cycles. A rate generator is often used to provide the precise time base required for accurate data acquisition.

The digital counters available in most DA&C systems accept TTL level signals and can be used to accumulate the number of input pulses. Counting can be started from a defined initial value and the counter can be configured to automatically reset to this value after it has been read. Internally, the counter actually decrements or subtracts a count for each input pulse. However, software can easily interpret the counter's data as a sum or difference from an arbitrary starting point. When a 16-bit counter exceeds 65535 or the initial count value (which ever is smaller) an overflow occurs generating a digital output. This signal can be used to activate external events. Of great significance is that the next input pulses simply cause the counter to decrement from 65535. Thus, if overflows are detected and accounted for, total counts of any size can be accumulated in the software.

Frequency measurements using counters can be accomplished in different ways depending upon the application. When the unknown frequency is a TTL signal, it can be applied directly to the counter circuit. Analog signals with an amplitude of at least 100mV can be converted to TTL levels with the PCI-20000's Trigger/Alarm module (PCI-20020M-1). Voltage dividers using resistors and/or zener diodes or opto-isolators can be used to scale down high-level signals. When using any kind of signal conditioning before the counter input, consideration should be given to possible resulting speed limitations.

Two distinct options exist for measuring high or low frequencies. The first method counts a known clock generator for the period of the unknown input signal. This provides high resolution for low-frequency signals, while minimizing the time required for the measurement. Generally this is used for frequencies below 10Hz. The second method counts cycles of the unknown input signal for a fixed time interval. The advantage of this technique is that it allows measurements beyond 8MHz. It is easy to implement an auto-ranging algorithm that optimizes resolution over a very wide frequency range.

Section 5
Software Techniques

Section 3 briefly discussed the role of DOS and other Operating Systems. Everyone knows that a computer is made to perform a useful task by "programming" it with a series of instructions. At the chip level, the system can respond only to the most primitive digital commands: an input is High or Low (On or Off, 1 or 0). Semiconductor circuits do not understand "Add", "Multiply", "Read" or "Print". They can understand that a given voltage is, or is not, present at an input. So ultimately, all communications with a computer is in terms of digital ones and zeros. In the beginning, only this "Machine Language" existed. Programming was slow, error prone (even more so than now!), difficult to maintain, and was approached only by specialists.

The invention of "Compilers" changed everything for the better. A compiler accepts (understands) alphanumeric inputs and translates them into machine-readable code. Sensible combinations of alphanumeric inputs (acronyms or other recognizable words) are defined to be equivalent to a pattern of digital inputs. Thus, the compiler (an intermediate program) provides a more practical human interface. Assembly language is the first step above machine code. An "assembler" is a low-level compiler that converts assembly language into machine code. The resulting code works with DOS and BIOS (defined in section 3) to further simplify a specific programming task. Be assured, that for most of us, Assembly language is not simple. It is however, much more manageable than machine code.

What else can be done to improve the situation? Languages such as BASIC can now come into play. These "High-Level" languages perform still more complex operations while presenting an additional degree of recognizable English (still dominated by jargon and special syntax, however). Each of the many high-level languages has been fine-tuned to excell in particular areas. BASIC is best known for the ease with which it can be learned and used. Other languages (C, PASCAL, etc.) are recognized for their execution speed and program maintainability.

As was explained earlier, the use of a high-level language requires that a compiler be used to translate the written program into machine code. So called "Compiled" languages are converted before run-time and executed in that form (i.e., C and PASCAL). The entire program must be debugged as a whole. "Interpreted" languages are "incrementally compiled" (i.e., BASIC). That is, as *each line* of the original code is read, it is then converted and executed. This permits a single program line to be written and tested independently.

One sometimes gets the feeling that the differences between computer languages are smaller than they are made out to be. Imagine asking an international gathering to select the best, single, spoken language for all to use! Similar disagreements about the "best" computer language also exist.

Software makes the computer-based data acquisition and control system operational. A low-cost, powerful hardware system is of little value without appropriate software.

Personal-computer-based data acquisition systems have been designed so that users have the opportunity to write specialized programs for data acquisition, storage, display, logging, and control in high-level languages. When software is provided with a data acquisition system, it should make these tasks as simple as possible for the user.

Three classes of software are generally available for PC based DA&C systems: tutorial and program development tools, function subroutine libraries and complete "turn-key" applications packages. Often the turn-key packages are menu-driven.

Program development tools and function libraries are packages designed to allow users to write their own unique applications software. They usually include "drivers" that provide the interface to the I/O hardware. These packages make it very easy to write programs in high-level languages such as BASIC, C, TURBO-PASCAL and ASYST. This type of programming is very flexible and is useful for general purposes.

Complete applications packages are designed to get the system going immediately, usually with no programming required. However, some of these packages offer users the facilities to enhance or modify the software to meet their own needs. Normally, this type of product is directed at a specific type of application. As a result these packages are often quite structured and less flexible than are the other classes of software.

Third-party software is that vast collection of "generic" software products designed by independent companies to serve hardware built by others. Some well known products include LABTECH Notebook, ONSPEC and LOTUS 1-2-3. These and many other software packages provide data collection, analysis, plotting, and control capabilities. In Section 9 of this handbook are several examples of third-party software being used with PCI products. Section 11 lists additional programs.

Many, if not most, data acquisition and control applications depend upon the timely execution of read/write operations. When speed and/or timing are critical, three techniques for software control should be considered: "polling", "interrupts" and "direct memory access" (DMA).

As would be expected, each has its special merits and requirements. Polling is the simplest method for detecting a unique condition and then taking action. This involves a software loop that contains all of the required measurement, analysis, decision-making algorithms and planned actions. The data acquisition program periodically tests the system's clock or external trigger input to sense a transition. Whenever a transition occurs, the program then samples each of the inputs and stores their values in a "frame." A frame is simply a list that contains the values representing the specified inputs at a given time. The frames can be stored in RAM, disk or other types of

memory. Each time the program senses a clock "tick", the inputs are scanned and converted, and a new frame is added to memory. In this mode, the IBM PC can support a data acquisition rate of about 54kHz, while the PC/AT can support about 89kHz. On the other hand, the design of the PC is such that potentially significant variations (or jitter) in timing can occur. In the IBM PC, jitter of approximately 12 microseconds is not uncommon. In addition, the PC is continuously busy when the polling loop is operational, and hence no other tasks can be serviced. When an application can not tolerate these characteristics, interrupt techniques may be indicated.

Interrupts do provide a means of tightly controlling the timing of events, while allowing the processing of more than one task. Multitasking systems are also known as "foreground/background" systems. One way of putting data acquisition in the background, is to relegate it to an interrupt routine. The clock or external timing signal, rather than being polled continuously, is used to generate an interrupt to the computer. Whenever the interrupt occurs, the computer suspends current activity, and executes an "interrupt service routine." The interrupt service routine in this case might be a short program which acquires one frame of data, and stores it in memory. The computer can perform other operations in the foreground while collecting data in the background. Whenever a clock tick or external interrupt occurs, the computer will automatically stop the foreground processing, acquire the data, and then resume where it left off. The reaction speed of the interrupt system is much slower than that of a well written polling loop. This results because the interrupt mechanism in most computers involves a significant amount of software overhead. Speed, for an IBM PC, is about 4kHz in the interrupt mode. Also, the software complexity of interrupts can be significant. In most cases the programmer must be prepared to write a substantial amount of assembly language code. In contrast, most polled systems can be written in a high-level language. Interrupts are useful in situations where the acquisition rate is slow, timing accuracy is not a priority, and background operation is important. When the amount of time required to service an interrupt is small, compared to the rate at which the interrupts can occur, then this technique yields excellent results. These factors should make it clear that careful thought is warranted before making a polling/interrupt question.

DMA is the hardware/software technique that allows the highest speed transfer of data, to or from random access memory (RAM). Given the potentially more expensive hardware, DMA can provide the means to read or write data at precise times without restricting the microprocessor's tasks. For example, the PCI-20000 system, under DMA control, can read or write any combination of analog, digital or counter/timer data to or from RAM at 360K bytes/second. This is accomplished by taking minimal time from the other tasks of the microprocessor. The amount of time required to respond to a DMA request is very small compared to the time required to service an interrupt. This makes the goal of foreground/background operation, at high speed, possible. For additional information about DMA, please refer to Section 3 of this handbook.

Section 6
Field Signals & Transducers

A good DA&C system does everything practical to simplify the handling of a wide variety of field signals. However, it is helpful for the user to have a basic knowledge of the types of signals that the system may be called upon to read or generate. This section reviews what the most common signals consist of, and shows how the DA&C system deals with them.

Transducers - Whatever the phenomena detected or the device controlled, "transducers" play a vital role in the DA&C system. It is the transducer that makes the transition between the physical and the electrical world. Remember that data acquisition and control can involve both input and output signals. Input signals can represent force, temperature, flow, displacement, count, speed, level, pH, light intensity, etc... Output signals can control valves, relays, lamps, horns, motors, etc... The electrical equivalents produced by input transducers are most commonly in the form of voltage, current, charge, resistance or capacitance. As shown later, the process of "Signal Conditioning" will further convert these basic signals into voltage signals. This is important because the major interior blocks of the DA&C system can only deal with voltage signals.

Signal Types - It is necessary to further define three types of voltage signals: analog, digital and pulse. While all signals are assumed to be changing with time, analog signals are the only ones to have information contained in their incremental variations in amplitude. The pulse signals referred to here are similar to the digital signals in many respects. Both digital and pulse signals are of uniform amplitude, and are represented by only two possible values (high and low). Typically, these high and low levels are approximately 5 and 0 volts respectively (TTL levels). The actual allowable ranges for TTL signals are:

low level = 0V to 0.8V,
high level = 2.0V to 5.0V

However, other levels including 110 or 220 VAC (line voltage), can be accommodated.

So, with analog it is important "how" high the signal is, while with digital it matters only "if" the signal is high or low (on or off, true or false). The distinction between digital and pulse signals lies in the information conveyed and the type of data acquisition hardware utilized. Digital signals are sometimes called "discrete" signals. A given digital "bit" is one channel of the DA&C system's digital port. While all digital signals have the potential to be changing states at high speed, information is usually contained in the static state of a bit or group of bits, at a given time. In contrast, pulse information is usually contained in the number of state transitions that have occurred or in the rate at which the state transitions are occurring (pulses/second). Refer to Figure 6.1 for a look at the differences between analog and digital signals.

Analog signals will be transformed into a digital representation (binary number) by the system's analog-to-digital converter (A/D). When analog outputs are required, they will be generated by the system's digital-to-analog converters (D/A). Analog inputs usually come from some type of preamplifier where the primary sensor signal has been conditioned and amplified for presentation to the data acquisition system. Most preconditioned signals are of a relatively high amplitude, in the range of ±1 to ±10 Volts. However, many primary sensors, such as thermocouples, photovoltaic cells, piezoelectric sensors and biomedical sensors produce small signals that may have a full scale range of only 10 millivolts. A quality data acquisition system must handle both high- and low-level signals with equal ease and accuracy.

Thermocouples - The thermocouple (TC) is so common for temperature measurement in industry and science, that it will be given special treatment. Physically, a TC is a junction of two dissimilar metals. This junction produces a thermal EMF proportional to the temperature of the junction (Seebeck effect). Temperatures of -200 °C to +4000 °C can be measured. The output voltage is usually in the range of -10 to +50 millivolts and has an average sensitivity of 10 to 50 microvolts/°C, depending upon the TC used. Many thermocouple types, using different combinations of metal alloys, are in wide use. For convenience, alphabetic letter designations have been given to the most common.

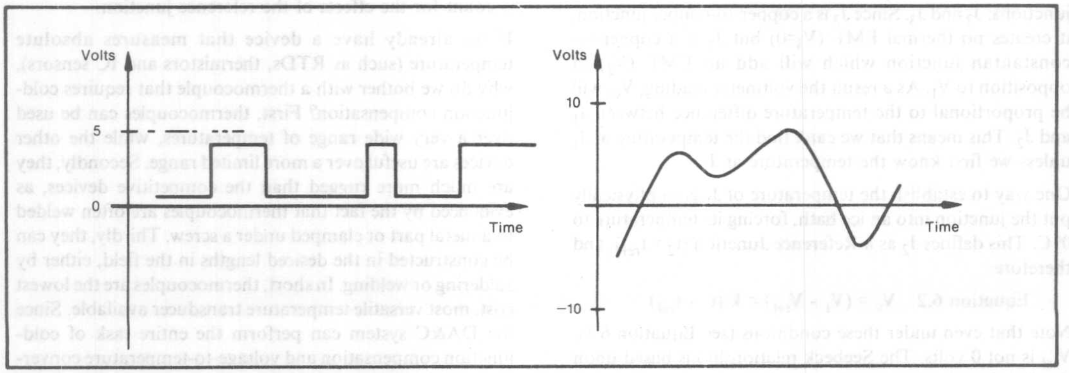


FIGURE 6.1. Digital and Analog Signals.

These include:

J	Iron	- Constantan	(Fe-C)
K	Chromel	- Alumel	(Ch-Al)
T	Copper	- Constantan	(Cu-C)

Tungsten, rhodium and platinum are also popular metals, particularly at very high temperatures.

While TCs are both low in cost and very rugged, they are not without their limitations and applications problems. Accuracy is generally limited to 1-3% due to material and manufacturing variations. In addition, response time is slow (on the order of several seconds) and both nonlinearity and multiple junction phenomena must be compensated.

The Law of the Junction - A single thermocouple junction generates a voltage proportional to temperature:

Equation 6.1 $V = k\ (t)$,

where "k" is the Seebeck coefficient, defining a particular metal-metal junction and "t" is in degrees Kelvin.

We cannot measure this Seebeck voltage directly. When we connect the TC to a measuring system the connection leads themselves create a new thermoelectric circuit. As an example, let us connect a voltmeter to a copper-constantan (Type T) thermocouple as in Figure 6.2.

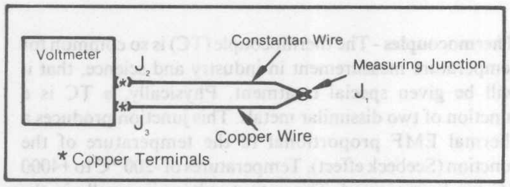


FIGURE 6.2. The Thermocouple Measurement Problem—Extra Junctions.

We would like the voltmeter to read only V_1 (of J_1) but by connecting the voltmeter in an attempt to measure the output of Junction J_1, we have created two more metallic junctions: J_2 and J_3. Since J_3 is a copper-to-copper junction, it creates no thermal EMF (V_3=0) but J_2 is a copper-to-constantan junction which will add an EMF (V_2) in opposition to V_1. As a result the voltmeter reading, V_v, will be proportional to the temperature difference between J_1 and J_2. This means that we can't find the temperature at J_1 unless we first know the temperature at J_2.

One way to establish the temperature of J_2 is to physically put the junction into an ice bath, forcing its temperature to 0°C. This defines J_2 as a Reference Junction ($t_2 = t_{ref}$), and therefore:

Equation 6.2 $V_v = (V_1 - V_{ref}) = k\ (t_1 - t_{ref})$

Note that even under these conditions (see Equation 6.1), V_{ref} is not 0 volts. The Seebeck relationship is based upon the Kelvin (absolute zero) scale. It is also important to remember that k is highly nonlinear with respect to temperature. However, we are fortunate that tables have been compiled that yield not only V_{ref} at any temperature, but more directly, V_v when t_{ref} is 0°C. These tables take variations in k into account and can provide t_1 directly in terms of V_v, assuming that t_{ref} is at 0°C. Note that both the measuring and reference junctions are both Cu-C.

As we have seen in this example, a copper-constantan TC is a special case because the copper wire is the same metal as the voltmeter terminals. It is interesting to look at a more general example using iron-constantan (Type J). The iron wire increases the number of dissimilar metal junctions in the circuit, as J_3 becomes a Cu-Fe thermocouple junction. However, it can be shown that if the Cu-Fe and the Cu-C junctions (at the termination panel) are at the same temperature, the resulting EMF is equivalent to a single Fe-C junction. This allows us to again use Equation 6.2, noting that both the measuring and reference junctions are of the same materials (Fe-C, in this case). Again, it is very important that both "parasitic" junctions be held at the same reference temperature. This can be aided by making all connections on an isothermal (same temperature) block.

Clearly, the requirement of an ice bath is undesirable for many practical reasons. Taking our analysis to the next logical step, we are reminded that Equation 6.2 does not require that t_{ref} be at any special temperature. It is only required that the reference temperature be accurately known. If we can measure the temperature of the isothermal block (the reference junction), independently, we can use this information to compute the unknown temperature, t_1. Devices like thermistors, RTD's, and semiconductor sensors provide us with a way to measure the absolute temperature of the reference junction. Therefore, under computer control, we simply:

1) Measure t_{ref} and compute the equivalent TC voltages for the parasitic junctions (V_{ref}).
2) Measure V_V and subtract V_{ref} to find V_1.
3) Convert V_1 to the desired temperature t_1.

This procedure is known as software "cold-junction" compensation, because it relies upon the computer to account for the effects of the reference junction.

If we already have a device that measures absolute temperature (such as RTDs, thermistors and IC sensors), why do we bother with a thermocouple that requires cold-junction compensation? First, thermocouples can be used over a very wide range of temperatures, while the other devices are useful over a more limited range. Secondly, they are much more rugged than the competitive devices, as evidenced by the fact that thermocouples are often welded to a metal part or clamped under a screw. Thirdly, they can be constructed in the desired lengths in the field, either by soldering or welding. In short, thermocouples are the lowest cost, most versatile temperature transducer available. Since the DA&C system can perform the entire task of cold-junction compensation and voltage-to-temperature conversion, using a thermocouple becomes as easy as connecting a pair of wires.

When selecting a thermocouple the following factors should be taken into account:

Type J **Lowest cost**, Highest sensitivity, Moderate accuracy. Should not be used above 760°C because of severe decalibration.

Type K Moderate cost, Moderate sensitivity, Low accuracy, **High temperature range**. Can be used to 1372°C due to its high resistance to oxidation.

Type T Moderate cost, Moderate sensitivity, High accuracy. **Very useful at low temperatures.** Because one lead is copper, cold junction compensation is not required when making differential temperature measurements with two back-to-back TCs.

For J, K and T type thermocouples, the "red" colored lead is always the negative terminal.

Thermistors - The thermistor is a metal oxide or semiconductor device that changes resistance with temperature. While positive temperature coefficient devices are available, most units exhibit a negative slope. This temperature coefficient can be as large as several percent per degree Celsius. This makes it possible to resolve smaller changes than with other devices (0.01°C). The accuracy of thermistors is typically 10 times better than that of thermocouples, yielding ±0.1°C under some conditions. Only the Platinum RTD has better accuracy. The physically small size and high nominal resistance are significant advantages. Small size yields a fast response while the high resistance makes any error, due to lead-wire resistance, small.

Along with the high sensitivity goes a high degree of nonlinearity. However, several manufacturers offer devices that have excellent conformance to published tables. While an individual unit exhibits a third-order logarithmic relationship, combinations of positive and negative slope devices can be made to have highly linear relationships. These units can be used from -50 to +100°C. In addition to the limited temperature range, attention must be given to the fragile nature of these devices. Careful mounting and handling must be used to avoid accuracy-destroying stress or catastrophic crushing.

Since it is basically a resistor, a thermistor can be read in several ways. These include current excitation (read a voltage) and voltage excitation (a voltage divider is formed with a fixed resistor). In either case, current must be passed through the measuring device. This will generate internal power dissipation that can produce an error-causing temperature rise. As a general rule, the self-heat error associated with this device can be estimated by dividing the proposed internal power dissipation by 8 milliwatts (yielding rise in °C). This rule applies to small bead thermistors in a conductive environment (like oil or water). In all cases, excitation levels must be held to a very low level to achieve high accuracy.

Resistance Temperature Detectors (RTDs) - As Thermistors, RTDs exhibit a changing resistance with temperature. Several different metals can be used to produce RTDs, but for a number of reasons, platinum has proven to be the most widely used. One notable exception to this rule is tungsten, which does find applications at very high temperatures. RTDs always have a positive temperature coefficient, with a small nonlinearity. For accurate measurements a third-order polynomial correction should be applied. Many data acquisition systems provide this built-in linearization capability.

Most RTDs are of either wire-wound or metal-film designs. The film design offers faster response time, lower cost and higher resistance values than the wire-wound types. The more massive wire-wound designs are more stable with time. High resistance is desirable because of the potential for lead-wire induced errors. However, even the so-called high resistance units require careful attention to lead-wire effects. Because of the excitation current required to produce a measurable signal, self-heating can also be a factor. However, the dissipation constant of an RTD is about ten times that of a thermistor. In this case, an estimate of the temperature rise (in °C) can be found by dividing the internal power dissipation by 80 milliwatts. Again, this is a general rule that applies to small RTD's in a conductive fluid like oil or water.

Most platinum RTDs are built with 100 or 200 ohm elements. 100 ohm metal-film devices seem to be the most popular. These units have sensitivities of about +0.4 ohms/°C. The combination of low sensor resistance and low sensitivity suggests the use of a bridge type of measuring configuration.

Solid State Temperature Sensors - These devices are derived from modern silicon integrated circuit technology, and are often referred to as Si sensors. They consist of electronic circuits that exploit the temperature characteristics of active semiconductor junctions. Versions are available with either current or voltage outputs. In both cases the outputs are directly proportional to temperature. Not only is the output linear but it is of a relatively high level, making the interpretation very easy. The most common type generates 1 microamp per degree Kelvin (298 microamps at 25°C). This can be externally converted to a voltage by using a known resistor. The usable temperature range is -50 to +150°C. The stability and accuracy of these devices are good enough to provide readings within ±0.5°C. It is easy to obtain 0.1°C resolution.

Strain, Pressure, Force, Position, Displacement and Level - These and many other types of transducers are often characterized by their responses to physical movement. Crystal and resistive strain gages, linear voltage-displacement transducers (LVDTs), slidewires (resistive potentiometers), and capacitive sensors are among the most common. While each of these sensors is based upon very different principles, the ultimate output signals are ordinary voltages, currents or impedances. These signals are directly or indirectly represented by analog voltage levels. Hence, the techniques described in this handbook can be applied to these types of transducers. Sensors that require external excitation present an accuracy dilemma. Invariably, higher excitation levels yield greater transducer output. However, this also leads to internal power dissipation that can cause errors even in mechanical devices. An optimum excitation

level exists for each type of device. If additional information about a specific transducer is desired, reference to the manufacturer's data sheet or to one of the available texts is suggested.

Flow, Speed and Count - Flow and speed can be measured in several ways. These include resistive, piezoelectric and thermal techniques. As discussed previously, these methods ultimately generate analog voltage signals. Transducers such as shaft encoders, paddle wheels (turbine), and both optical and magnetic pickups typically have digital or pulse outputs. The desired speed, rate or number of events can be determined by using digital counting or frequency measurement techniques. The methods for acquiring analog, pulse and frequency signals are covered elsewhere in this handbook.

Light Intensity and Chemical Action - These parameters are often encountered in density, spectroscopy and pH measurements. The transducers are characterized as having very high output impedance. The light-activated devices typically are modeled as current sources, while the chemical devices look like voltage sources with high series resistance. In most cases, the raw signals from these types of transducers can not be directly processed by standard DA&C systems. Even the excellent characteristics of the modern PGAs discussed earlier are inadequate in these specialized applications. However, many transducer manufacturers include the necessary preprocessing as an integral part of their product. When they do, the signals are then high-level voltages or currents which can be read as outlined elsewhere in this handbook. (Section 4.)

Low-level currents are often preprocessed with a FET input op-amp operating as a current-to-voltage converter (transimpedance amplifier). As suggested earlier, all amplifiers have finite input bias currents that can produce errors. FET input amplifiers have bias currents that are often below 1 picoamp (10 to 100 femtoamp units are available), which makes them useful for the majority of practical applications.

High impedance voltage sources are also preamplified with FET input amplifiers. In this case the op-amp is configured as a non-inverting voltage amplifier. This can yield input impedances on the order of 10^{14} ohms, which satisfies most high impedance transducer applications.

Detailed information about selecting and applying FET amplifiers can be found in the Burr-Brown components data book, as well as in several Burr-Brown text books and application notes.

Resistance - Resistance signals arrive at the data acquisition system from primary sensors such as strain gages and RTDs. A resistance sensor is usually measured as part of a Wheatstone "bridge" circuit. A bridge is a symmetrical, four-element, circuit that enhances the system's ability to detect small changes in the sensor. The sensor can occupy 1, 2 or 4 arms of the bridge, with any remaining arms being filled with fixed resistors. A differential voltage signal is developed across the arms of the bridge when the sensing resistors vary from their nominal values as a result of temperature or strain.

Transducer excitation, as well as provisions for the insertion of bridge-completion components, can be provided on signal "termination panels." While both voltage and current excitation can be used, current excitation is generally more desirable. This is because current excitation provides a more linear output response, making the data interpretation easier. The PCI-20000 system provides adjustable current sources that can be optimized for the type of transducer being measured.

The diagrams below show some of the more common configurations for resistive bridge elements. Bridge-completion resistors should be of very high precision (typically 0.05%). Stability is actually the most important characteristic of the bridge-completion elements. Initial inaccuracies can be calibrated out, but instability always appears as an error. More information about bridge circuits and their transfer functions can be found in the Burr-Brown textbook series.

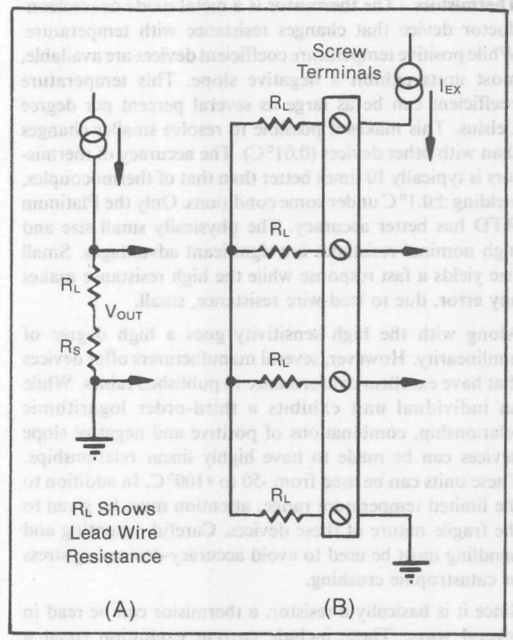


FIGURE 6.3. Measuring a Resistive Device.

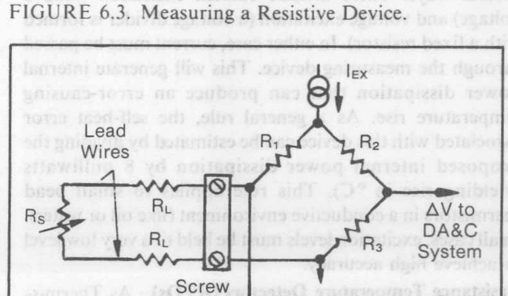


FIGURE 6.4. A Two-Wire Bridge Configuration.

Transducers, such as strain gages and RTDs, have relatively low sensitivity. That is, the change in resistance is small for a given change in the input parameter. Simply measuring the change in voltage (due to a current excitation) across the device is difficult. Not only is the change in voltage small, but it is "riding" on the device's quiescent (I · R voltage). The quiescent voltage greatly limits the amplification that can be used to amplify the voltage change. Let us explore this concept in more detail. Fundamentally, a DA&C system can measure only voltage. Fortunately, as has been suggested, all other types of signals can be transformed into voltages. To convert a changing resistance, we need only to "excite" it with a current. The voltage across the resistance is then (I · R). Figure 6.3A shows the basic idea.

A real application might involve a 100-ohm platinum RTD. To control internal self heating, the excitation level is usually limited to 2mA. Given that the sensitivity of this type of device is about +0.4 ohms/°C, the output will be about 0.8mV/°C. This is indeed a small signal that will require amplification. It would be useful to multiply the signal by 100 to 1000 times to make best use of the A/D's full-scale range (typically, 5 or 10V). However, the quiescent voltage across the RTD is (2mA · 100 ohms) = .2V. This limits the maximum gain to 10. Thus, in a 12-bit system, the smallest detectable temperature change is about 0.5°C. In contrast, the bridge circuits shown balance out the fixed or quiescent voltage drop, allowing greater magnification of the difference signal. This allows the detection of changes as small as 0.005°C.

The effects of lead-wire resistance should also be considered. The output voltage is proportional to the sum of the RTD resistance and the connecting wire resistance (R_L). In many applications this "wire error" can be very significant. Figure 6.3B suggests a solution. This is the so-called Kelvin or four-terminal connection. Wire resistance cannot be eliminated, but this measurement technique greatly reduces the effects. The idea is to connect two wires to each end of the measuring device. One lead carries the excitation current and the other senses the terminal voltage. Current in the sense or measurement lead is very small and can be assumed to be zero. This is because the DA&C system has a very high input impedance. Thus, no voltage drop occurs in the sensing lines. Note that under these conditions the drops in the excitation lines are not in the measurement circuit.

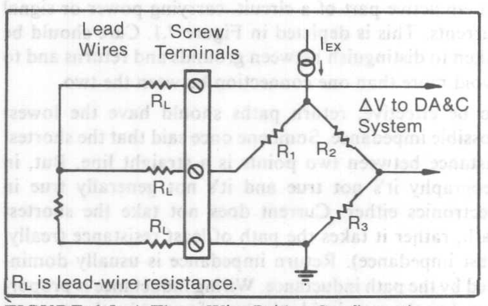


FIGURE 6.5. A Three-Wire Bridge Configuration.

The most commmon resistive sensors are of the one- and four-element types. Both the two-wire and three-wire bridge configurations are intended to monitor single-element transducers. These are transducers that are represented by a single resistor that is exposed to, or is sensitive to, the measured parameter. In the four-element circuit, all four of the resistors react to the measured parameter. As might be expected, this configuration offers four times the sensitivity of a single-element bridge. In addition, the four-element bridge offers the most linear response.

The two-wire bridge is very simple, but has the potential to be adversely affected by the series resistance of the connecting wires. As can be seen in Figure 6.4, the lead-wire resistance is indistinguishable from the transducer's resistance. Hence, this circuit is not usually employed in precision applications.

While the three-wire bridge requires an additional wire to be run to the sensor, several very important advantages are gained. If we make the reasonable assumption that the two wires bringing current to the sensor are of the same material and length, many of the potential error terms cancel. In Figure 6.5, it can be visualized that one lead resistance is in the top arm of the bridge while the other lead resistance is in the lower arm. The result is that most of the lead-wire effects are cancelled. However, when long leads (generally, over 10 feet) are used or the highest possible precision is desired, software correction of the lead effects can be employed. The resistance of the sense wire is of little significance because the current that flows in this lead is very small. Owing to the lead-wire error cancellation and the available computational power of the PC, this configuration is ideal for most DA&C applications.

Figure 6.6 shows a four-element bridge circuit. Here, the connecting lead wires do not introduce any significant error terms. Both of the power supply connecting wires are in series with a current source and hence do not affect the excitation level. A complete bridge does not have any connecting wires in series with the individual sensing resistors. As suggested above, this circuit has the highest sensitivity, the best linearity, and does not require bridge-completion resistors. Unfortunately, the complex manufacturing process for this type of sensor does result in relatively high cost.

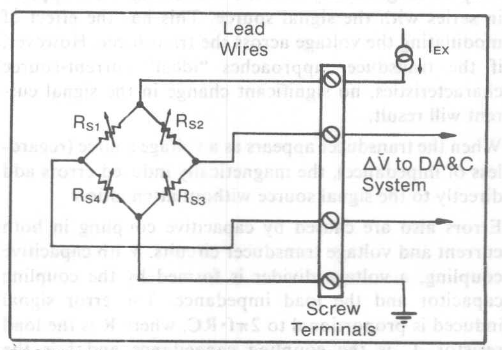


FIGURE 6.6. A Four-Wire/Element Bridge Configuration.

6

FIELD SIGNALS & TRANSDUCERS
BRIDGE CIRCUITS

Section 7
Wiring & Noise Considerations

It is a fact that signals entering a data acquisition system will include unwanted noise. Whether this noise is troublesome depends upon the signal-to-noise ratio and the specific application. In general, it is desirable to minimize noise to achieve high accuracy. Digital signals are relatively immune to noise because of their discrete (and high-level) nature. In contrast, analog signals are directly influenced by relatively low-level disturbances. The major noise transfer mechanisms include conductive, inductive (magnetic), and capacitive coupling. For example:

- The switching of high-current loads in nearby wiring can induce noise signals by magnetic coupling (transformer action).
- Signal wires running close to AC power cables can pick up 50/60 Hz noise by capacitive coupling.
- Allowing more than one power or signal return path can produce ground loops that inject errors by conduction.

Conduction involves current flow through ohmic paths (direct contact), rather than inductance or capacitance.

Interference via capacitive or magnetic mechanisms usually requires that the disturbing source be close to the affected circuit. At high frequencies, however, radiated emissions (electromagnetic signals) can be propagated over long distances.

In all cases, the noise level induced will depend upon the several factors that can be user influenced.

- Signal source "output impedance" (the transducer type).
- Signal source "load impedance" (the input impedance to the data acquisition system).
- Lead-wire; length, shielding and grounding.
- Proximity to noise source(s).
- Signal and noise amplitude.

Transducers that can be modeled by a current source are inherently less sensitive to magnetically induced noise pick-up than are voltage-driven devices. An error voltage coupled magnetically into the connecting wires appears in series with the signal source. This has the effect of modulating the voltage across the transducer. However, if the transducer approaches "ideal" current-source characteristics, no significant change in the signal current will result.

When the transducer appears as a voltage source (regardless of impedance), the magnetically induced errors add directly to the signal source without attenuation.

Errors also are caused by capacitive coupling in both current and voltage transducer circuits. With capacitive coupling, a voltage divider is formed by the coupling capacitor and the load impedance. The error signal induced is proportional to $2\pi f \cdot RC$, where R is the load resistor, C is the coupling capacitance and f is the interfering frequency. Clearly, the smaller the capacitance (or frequency) the smaller is the induced error voltage. However, reducing the resistance only improves voltage type transducer circuits.

Example:

> Assume that the interfering signal is a 110VAC, 60Hz power line, the equivalent coupling capacitance is 100pF and the terminating resistance is 250Ω (typical for a 4-20mA current loop). The resulting induced error voltage will be about 1mV, which is less than 1 LSB in a 12-bit, 10V system.
>
> If the load impedance were 100KΩ, as it could be in a voltage input application, the induced error could be much larger. The equivalent R seen by the interfering source depends upon not only the load impedance but also the source impedance and the distributed nature of the connecting wires. Under worst-case conditions, where the wire inductance separates the load and source impedances, the induced error could be as large as 0.4V. This represents about an 8% full-scale error.

Even though current type signals are usually converted to a voltage at the input to the data acquisition system, with a low value resistor, this does not improve noise performance. This is because both the noise and transducer signals are proportional to the same load impedance.

Before panic totally overwhelms the reader, it should be pointed out that this example does not take advantage of—or benefit from—shielding, grounding and filtering techniques.

Most noise problems can be solved by close attention to a few grounding and shielding principles:

- Do not confuse ground and return paths.
- Minimize wiring inductance.
- Minimize ground currents.
- Limit antennas.
- Maintain balanced networks where possible.

This sounds simple enough, but what is involved?

To begin with, redefine some common terms. A ground is NOT a signal or power supply return path. A ground wire connects equipment to earth for safety reasons, to prevent accidental contact with dangerous voltages. Ground lines do not normally carry current. Return lines are an active part of a circuit, carrying power or signal currents. This is depicted in Figure 7.1. Care should be taken to distinguish between grounds and returns and to avoid more than one connection between the two.

To be effective, return paths should have the lowest possible impedance. Someone once said that the shortest distance between two points is a straight line. But, in geography it's not true and it's not generally true in electronics either. Current does not take the shortest path, rather it takes the path of least resistance (really, least impedance). Return impedance is usually dominated by the path inductance. Wiring inductance is proportional to the area inside the loop formed by the current-

carrying path. Therefore, impedance is minimized by providing a return path that matches or overlaps the forward signal path. Note that this may not be the shortest or most direct route. This concept is fundamental to insuring proper system interconnections.

Three different grounding and connection techniques are suggested in Figure 7.2. The circuit in Figure 7.2A allows the signal return line to be grounded at each chassis. This may look like a good idea from a safety standpoint. However, if a difference in potential exists between the two grounds, a ground current must flow. This current multiplied by the wire impedance results in an error voltage, e_e. Thus, the voltage applied to the amplifier is not V_I, but $V_I + e_e$. This may be acceptable in those applications where the signal voltage is much greater than the difference in the ground potentials.

When the signal level is small and a significant difference in ground potentials exists, the connection in Figure 7.2B is more desirable. Note that the return wire is not grounded at the amplifier and ground current can not flow in the signal wires. Any difference in ground potential appears, to the amplifier, as a common-mode voltage (CMV). In most circuits the effects of CMV are very small, as long as the signal voltage plus CMV is less than 10V (10V is the linear range for most amplifiers). Additional information about common-mode rejection and single-ended versus differential amplifiers can be found in the handbook section "Data Conversion Principles."

If cost is not a limitation, Figure 7.2C offers the highest performance under all conditions. Injecting an isolator into the signal path faithfully conveys V_I to the amplifier while interrupting all direct paths. In this configuration multiple ground connections can be tolerated along with several hundred volts between the input and output circuits. Additional information on both analog and digital isolators can be found in the section on signal conditioning. Related product data sheets are included in the PCI-20000 system configuration and data sheets section.

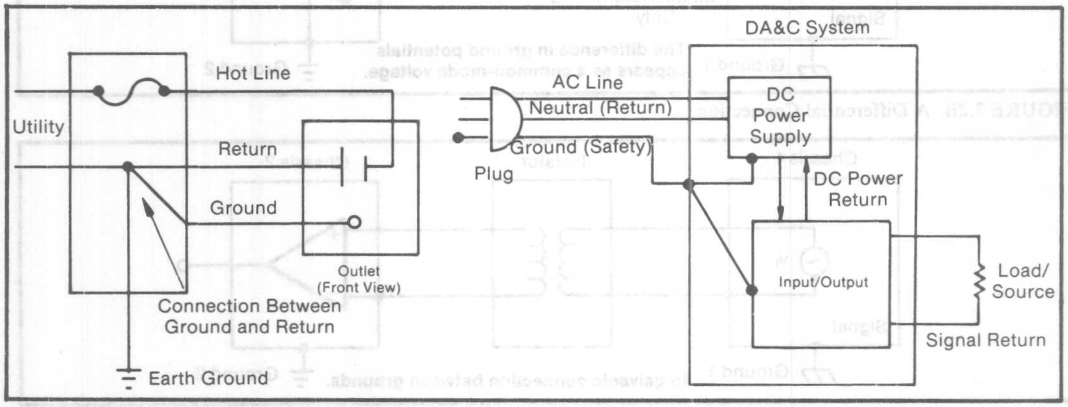


FIGURE 7.1. The Differences Between Ground and Return Conductors.

Cable Types—What kind of wire should be used to interconnect a system? We must first emphasize that a single piece of wire, is not generally useful. Circuits consist of complete paths, so we will refer to pairs of wires in this discussion. Basically, four kinds of wire are fundamental: the plain pair, shielded pair, twisted pair and coaxial cable. All but the coaxial (coax) wires are said to be balanced. Coax differs from the others in that the return line surrounds the central conductor. Technically, we should not call the outer conductor a shield because it carries signal current. It is significant that the forward and return path conductors do not model the same. In contrast a shielded pair is surrounded by a separate conductor (properly called a shield) that does not carry signal current.

Figure 7.3 suggests a simple model for a differential signal connection. The attributes of the signal source have been split to model the influence of a common-mode voltage (CMV). Focus on the effect of forward and return path symmetry in the cable. Assuming that the amplifier is perfect, it will respond only to the difference between V_a and V_b. Superposition allows us to analyze each half of the cable model separately and then to add the results. Z_1 is usually dominated by series inductance, while Z_2 is dominated by parallel capacitance. In any case, Z_1 and Z_2 form a voltage divider. If the dividers in both legs of the cable are identical, $V_a - V_b$ will not be influenced by CMV. If, however, the capacitance represented by Z_2 is different in the two paths, a differential voltage will result and the amplifier

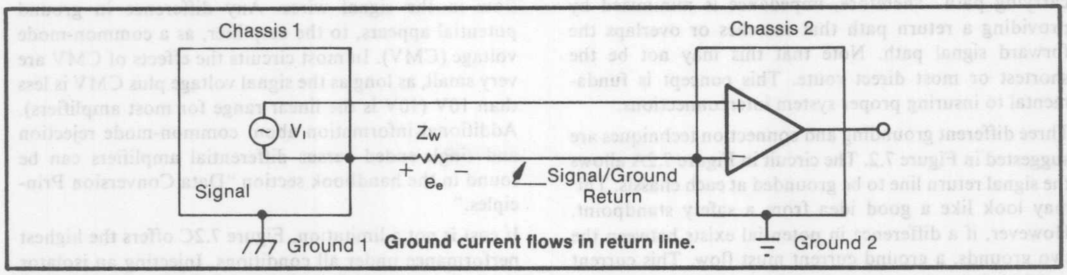


FIGURE 7.2A. A Single-Ended Connection.

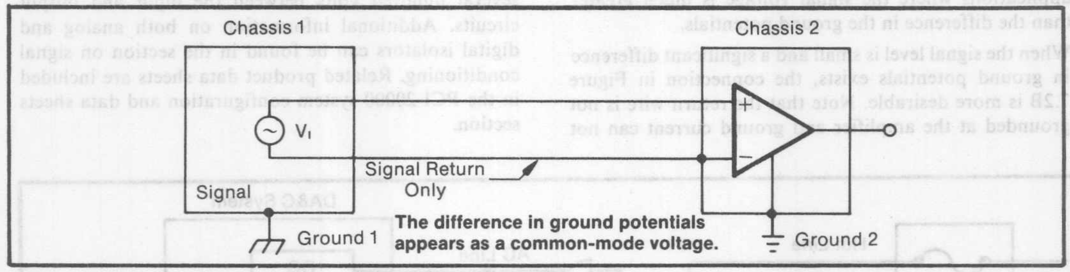


FIGURE 7.2B. A Differential Connection.

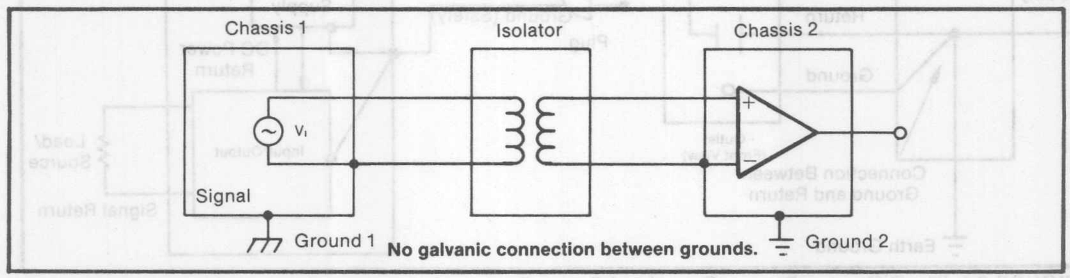


FIGURE7.2C. An Isolated Connection.

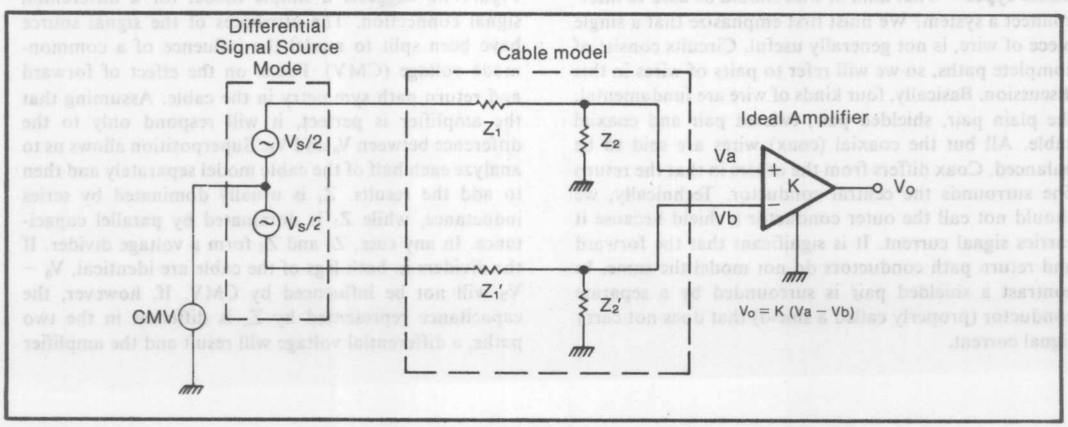


FIGURE 7.3. Influence of Cable Connectors on Common-Mode Signal Performance.

will be unable to distinguish the resulting CM error from a change in V_S.

Coax offers a very different capacitance between each of its conductors and ground. Not only does the outer conductor surround the inner, but it is also connected to ground. Thus, coax is intended for single-ended applications only. Note that even perfectly balanced cables can still attenuate differential signals.

Sometimes even a single-ended source is best measured with a differential amplifier. Refer again to Figure 7.2b. To maintain a high rejection of any ground difference potential, balanced cables are required.

One method of reducing errors, due to capacitive coupling, is to employ a "shield." Generally, there is little that can be done to reduce the actual capacitance (wire length and physical location are factors, however). Nevertheless, placing a conductive material (at ground potential) between the signal wires and the interference source is very useful. The shield blocks the interfering current and directs it to ground. Depending upon how "complete" the shield is, attenuations of more than 60dB are attainable. When using shielded wire, it is very important to connect only one end of the shield to ground. The connection should be made at the DA&C system end of the cable (input amplifier, etc.). Connecting both ends of the shield can generate significant error by inducing ground-loop currents.

A shield can work in three different ways:

- 'Bypassing' capacitively coupled electric fields
- 'Absorbing' magnetic fields
- 'Reflecting' radiated electromagnetic fields.

Another approach is to use twisted pairs. Twisted-pair cables offer several advantages. Twisting of the wires insures a homogenous distribution of capacitances. Both capacitances to ground and to extraneous sources are balanced. This is effective in reducing capacitive coupling while maintaining high common-mode rejection. From the perspective of both capacitive and magnetic interference, errors are induced equally into both wires. The result is a significant error cancellation.

The use of shielded and/or twisted-pair wire is suggested whenever low-level signals are involved. With low impedance sensors, the largest gage connecting wires that are practical should be used to reduce lead-wire resistance effects. On the other hand, large connecting wires that are physically near thermal sensing elements tend to carry heat away from the source, generating measurement errors. This is known as thermal shunting, and it can be very significant in some applications.

The previous discussion concentrated on cables making single interconnections. Multiconductor cables, for connecting several circuits, are also available in similar forms (i.e., twisted pairs, shielded pairs, etc.). Both round and flat (ribbon) cables are widely used. Because of the close proximity of the different pairs in a multiconductor cable, they are more susceptible to "crosstalk". Crosstalk is interference caused by the inadvertent coupling of internal signals via capacitive or inductive means.

Again, twisted pairs are very effective. Other methods include connecting alternate wires as return lines, running a ground plane under the conductors or using a full shield around the cable.

Still another noise source, not yet mentioned, is that of triboelectric induction. This refers to the generation of noise voltage due to friction. All commonly used insulators can produce a static discharge when moved across a dissimilar material. Fortunately, the effect is very slight in most cases. However, it should not be ignored as a possible source of noise when motion of the cables or vibration of the system is involved. Special low-noise cables are available that employ graphite lubricants between the inner sufaces to reduce friction.

The key to designing low-noise circuits is recognizing potential interference sources and taking appropriate preventive measures. Figure 7.4 can be useful when troubleshooting an existing system.

After proper wiring, shielding and grounding techniques have been applied, input filtering can be used to further improve the signal-to-noise ratio. However, filtering should never be relied upon as a fix for improper wiring or installation.

Cable Length Guidelines—What is the maximum allowable cable length? There is no direct answer to this question. The number of factors relating to this subject are overwhelming. Signal source type, signal level, cable type, noise source type(s), noise intensity, distance between the cable and the noise source(s), noise frequency, signal frequency range and required accuracy are just some of the variables to consider. However, experience has given us a "feel" for what often works. For example:

Analog, Current Source Type Signals -

Given: 4-20mA signal, Shielded wire, Bandwidth limited to 10 Hz, Accuracy required is 0.5%. "Average", industrial noise levels.

Cable Lengths of 1000 to 5000 feet (300 to 1500 meters) have been used successfully.

Analog, Voltage Source Type Signals -

±1 to ±10 volt signal, Shielded wire, Bandwidth limited to 10Hz, Accuracy required is 0.5%. "Average", industrial noise levels.

Cable Lengths of 50 to 300 feet (15 to 90 meters) have been used successfully.

Analog, Voltage Source Type Signals -

Given: ±10mV to 1volt signal, Shielded wire, Bandwidth limited to 10Hz, Accuracy required is 0.5%. "Average", industrial noise levels.

Cable Lengths of 5 to 100 feet (1.5 to 30 meters) have been used successfully.

OBSERVATION	SUSPECT	POSSIBLE SOLUTION	NOTES
Noise a function of cable location	Capacitive coupling	Use shielded or twisted pair	A
	Inductive coupling	Reduce loop area, use twisted pair or metal shield	B
Average value of noise— Is not zero	Conductive paths or ground loops.	Faulty cable or other leakage. Eliminate multiple ground connections.	C
Is zero	Capacitive coupling	Use shielded or twisted pair	A
Shield inserted— Ground significant	Capacitive coupling	Use shielded or twisted pair	A
Ground insignificant	Inductive coupling	Reduce loop area, use twisted pair or metal shield	B
Increasing load— Reduces error	Capacitive coupling	Use shielded or twisted pair	A
Increases error	Inductive coupling	Reduce loop area, use twisted pair or metal shield	B
Dominant feature— Low frequency	60Hz AC line, motor, etc	(1) Use shielded or twisted pair. (2) Reduce loop area; use twisted pair or metal shield. (3) Faulty cable or other leakage; eliminate multiple ground connections.	
High frequency	Electromagnetic radiation	Complete shield	D
Noise a function of cable movement	Triboelectric effect	Rigid or lubricated cable	
Noise is "white" or 1/f	Electronic amp, etc.	Not a cable problem	

NOTES: (A) Connect shield to noise return point and check for floating shields.
(B) Nonferrous shields are good only at high frequencies. Use MuMetal shields at low frequencies.
(C) Could be capacitive coupling with parasitic rectification, i.e. nonlinear effects.
(D) Look for circuit elements whose size is on the order of the noise wavelength (antennas). Openings or cracks in the chassis or shields with a dimension bigger than the noise wavelength/20 should be eliminated.

FIGURE 7.4. Troubleshooting Guide for Noise.

Digital, TTL Type Signals

Ground-plane type cable, "Average", industrial noise levels.

Cable Lengths of 10 to 100 feet (3 to 30 meters) have been used successfully.

Ground-Plane cable reduces signal reflections, ringing and RFI. Special termination networks may be required to maintain signal integrity and minimize RFI. If "squaring circuits" (Schmidt triggers) are used to restore the attenuated high frequency signals, improved performance can be realized.

Remember, this information is offered as "typical" of what might be encountered. The actual length allowed in a particular application could be quite different.

The following relationships are offered as an aid to visualizing the influence of the most significant factors determining cable length. These relationships show **how** the various parameters affect cable length. These relationships are **not equations**, and will not allow the calculation of cable length.

For Current Source Type Signals:

Allowable length is proportional to $\dfrac{I_s \cdot D_n \cdot C_f}{f_n \cdot A \cdot N_i}$

For Voltage Source Type Signals:

Allowable length is proportional to $\dfrac{V_S \cdot D_n \cdot C_f}{f_n \cdot A \cdot N_i \cdot R_L}$

Where:

I_s or V_s equals the signal level,
C_f equals the coupling factor which is inversely proportional to the effectiveness of any shielding or twisting of the wires.
D_n equals the distance to the noise source,
f_n equals the noise frequency,
A equals the required accuracy,
N_i equals the noise source intensity, and
R_L equals the equivalent resistance to ground at the signal input.

Section 8
Signal Conditioning

We are all familiar with the old cliche "a chain is only as strong as its weakest link." It is also true that a DA&C system is only as good as its input signals. As suggested in Figure 1.1, a DA&C system is made up of several links. Usually, the total system is partitioned at the transducer/signal conditioning interface. That is, the transducer is considered to be the signal source. Signal conditioning can consist of several operations, including current-to-voltage conversion, scaling, filtering, isolating and amplifying. Optimum signal conditioning is important to maintain the highest accuracy. Signal conditioning can be physically installed at a number of locations, including at the signal source, at the input to the amplifier, at the A/D, etc. The most convenient and hence most common location is on signal termination panels. A large variety of standard PCI signal-conditioning termination panels are available. Both "passive" and "active" panels support a wide range of digital and analog applications. All panels provide for field wiring connections through convenient screw terminals. Passive panels are designed to accommodate most common R, L, C and diode arrangements. This allows the user to configure voltage dividers, filters, surge suppressors, etc. The active panels also include programmable differential amplification, bridge completion, bridge excitation, cold junction compensation and optional isolation. Shielded ribbon cables are used to bring the termination panel's outputs/inputs to the data acquisition electronics (board, module or box).

Surge Protection—When a system can be subjected to unintentional high-voltage inputs, it is required that protection be provided to avoid possible destruction of the equipment. High-voltage inputs can be induced from: lightning, magnetic fields, static electricity, and accidental contact with power lines, to name just a few sources.

In Figure 8.1, C1 and C2 are voltage surge suppression capacitors. In some applications they must be rated for high voltage because transients in power stations or other noisy environments can exceed 1000V. The values of C1 and C2 should be as large as physically possible, and these capacitors should be positioned as close as possible to the signal entry point of the system. Capacitors with low series impedance at high frequencies should be selected. This requirement eliminates electrolytic-type capacitors. If the input signal can change polarity, polarized capacitors must be avoided.

Components R1, R2 and M1, M2 form a voltage clamp to further insure that transients will not get to the inputs of the DA&C system. M1 and M2 are metal oxide varistors (MOVs). They are semiconductor devices that can react very quickly to absorb high energy spikes.

Consideration should be given to the possible leakage current of the MOVs. If the series R is large, the leakage could appear as a significant temperature-dependent offset voltage (I • R). A practical example follows:

$$R = 1k\Omega,\ 1/8W \qquad M = 10V\ MOV$$

The resulting offset voltage will be insignificant (less than 1LSB). If a steady state, high voltage fault should occur (i.e., contact with 120VAC, etc.), the MOV will protect the DA&C system while the resistor will act as a fuse. Good results have been recorded with the General Industries SCL10C TransZorb MOV.

Low Pass Filtering—By averaging a series of incoming signals, we effectively increase the signal-to-noise ratio. Given the speed and math capabilities of modern DA&C systems, averaging is readily employed. Averaging will be most effective in reducing the effects of random, non-periodic noise. It is less effective in dealing with 50/60Hz or other periodic noise sources. It is important to remember that all noise filtering techniques, whether hardware or software, are designed to filter specific types of noise. The PCI-20000 signal termination panels have provisions for the user to install one or two poles of passive filtering. Figure 8.2 shows an example of an effective, single-ended, double-pole circuit to attenuate 50/60Hz noise. The filter has a −6dB cutoff at about 1Hz while attenuating 60Hz about 52dB (380 times).

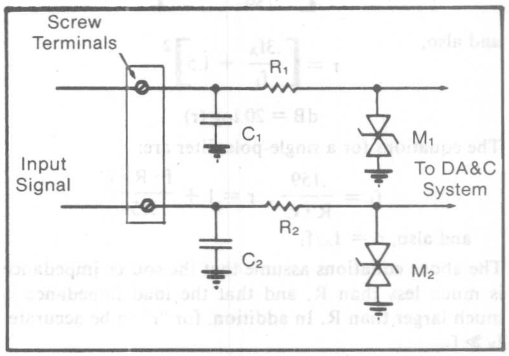


FIGURE 8.1. Differential-Input Surge Suppression Filtering.

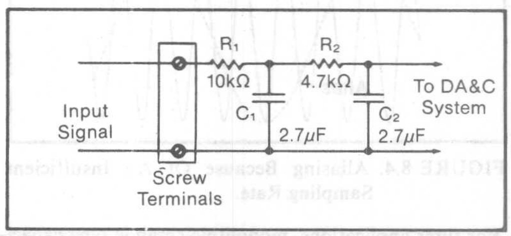


FIGURE 8.2. A 1Hz Low-Pass Filter (Single-Ended, 2-Pole).

Figure 8.3 suggests a differential, two-pole, low-pass filter. In contrast to the circuit in Figure 8.2, this can be used in balanced applications. Note that any mismatch of the attenuation in the top and bottom path will result in a degradation of the system common-mode rejection

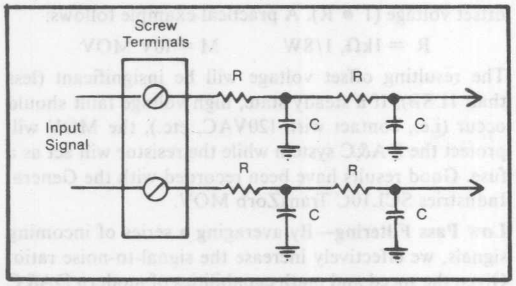


FIGURE 8.3. A Low-Pass Filter (Differential, Two-Pole).

ratio, CMRR (see Section 7). Therefore, the resistors and capacitors should be carefully matched to each other. If it is given that all of the resistors and capacitors are of equal values, the pole position (f_1) for this two-section filter is:

$$f_1 = 0.048/R \cdot C$$

and the approximate attenuation ratio, at a given frequency (f_X), V_{OUT}/V_{IN} (r) is:

$$r = \left[\frac{f \cdot R \cdot C}{.159} + 1.5\right]^2$$

and also,

$$r = \left[\frac{.3f_X}{f_1} + 1.5\right]^2$$

$$dB = 20 \log (r)$$

The equations for a single-pole filter are:

$$f_1 = \frac{.159}{R \cdot C}, \quad r = 1 + \frac{f \cdot R \cdot C}{.159}$$

and also, $r = f_X/f_1$

The above equations assume that the source impedance is much less than R, and that the load impedance is much larger than R. In addition, for "r" to be accurate, $f_X \gg f_1$.

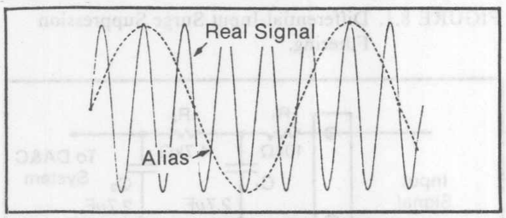


FIGURE 8.4. Aliasing Because Of An Insufficient Sampling Rate.

For filter applications, monolithic-ceramic type capacitors have been found to be very useful. They possess very high density (small in size for a given capacitance), are low leakage and are non-polar. Values up to $4.7\mu F$ at 50V are commonly available.

Filters have special significance in computerized DA&C systems beyond noise reduction. These devices are generally sampled-data systems. This implies that while data is recorded on a regular basis, it is not taken continuously. That is, there are gaps between the data points. Thus, when the data is interpreted, certain assumptions must be made about what the data is doing between the known points. In most cases it is assumed that the gaps contain information that falls on a straight line drawn between the known data points. This is known as linear interpolation. When this type of presumption is not sufficiently accurate, the logical recourse is to increase the sampling rate. This makes the gaps smaller by adding additional 'real' data points. Nyquist has provided us with a firm theoretical foundation upon which to deal with sampled data. In the simplest of terms, he states that a signal must be sampled at a minimum of two times the highest input frequency.

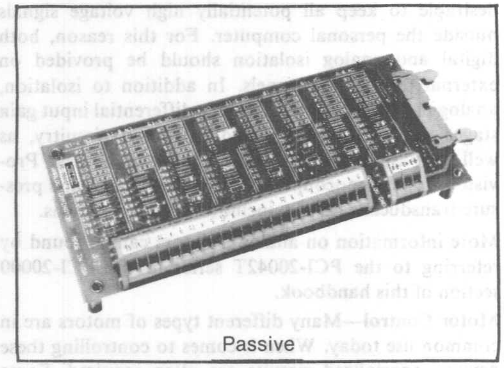
With pulse waveforms there can easily be very significant harmonics far beyond the repetition rate. Frequencies out to $1/t_r$ are often important (t_r is the pulse rise time). The danger of under-sampling is that erroneous conclusions can be drawn about the input signal. It is not simply a matter of overlooking something, but of reaching totally wrong conclusions about the basic makeup of the data. See Figure 8.4 for an example. Note that sampling a pure sine wave (containing only the fundamental frequency) at a rate in violation of the Nyquist criterion leads to meaningless results. In this example, it suggests the presence of a totally nonexistent signal. This phenomenon is known as aliasing.

Given that there are practical limits to the maximum sampling rate, another action must be taken to insure that the input data does not violate the Nyquist criterion. This involves the use of an input filter. In the ideal case, this filter would have infinite rejection beyond the cutoff frequency. This would allow the filter to be set at one half the sampling rate. Because perfect filters are not available, however, an appropriate compromise must be made between system bandwidth and sampling speed.

While high-order active filters are sometimes used to implement anti-aliasing networks, two-pole passive filters are usually sufficient. Depending upon the termination panel selected, accommodations for one, two or three pole filters can be provided.

In summary, filtering is intended to attenuate unavoidable noise and to limit bandwidth to comply with the Nyquist sampling theorem (the maximum signal frequency must be limited to half of the sampling frequency). It must always be remembered that filtering is not intended as a substitute for proper wiring techniques.

Current Conversion—The need to measure input current is quite common in DA&C systems. The outputs from remote sensors are often converted to high-level, 4 to 20mA signals at the source. This task is easily accomplished with a wide variety of transmitters. At the measurement system, current is easily converted back to a voltage with a simple resistor. Values of 250 to 500Ω are often used. The largest resistor that does not cause an over-range condition should be employed. This insures the maximum resolution. Stability of the resistor is essential. The exact value is not important. Most systems



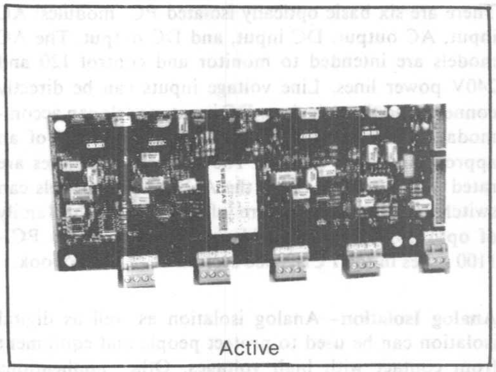


FIGURE 8.5. Examples of Passive and Active Termination Panels.

have software provisions to calibrate the measurement sensitivity for each channel at the time of installation. Low cost, 1%, metal-film resistors are usually adequate. The system's signal conditioning termination panels have provisions for mounting these devices.

Contact Sensing—When interfacing to relay or switch contacts, a "pull-up" current must be provided. The pull-up current converts the opening and closing of the contacts to TTL levels. Because all metal surfaces tend to oxidize with time, poor relay contacts can result. This oxidation can be cleaned away by passing a minimum current through the relay contacts. Both level generation and contact wetting can be accomplished by connecting resistors between the input lines and the +5V power supply. This can be implemented on the signal termination panel as shown in Figure 8.6. A value of 250Ω for R_1 will provide 20mA of wetting current, which is usually enough to keep most contacts free of oxide buildup. R2 and C1 function as a debounce filter to reduce erroneous inputs due to the mechanical bouncing of the contacts. When the switch is open, the system sees +5V. When the switch is closed, the input is 0V. This satisfies the TTL requirements of the system.

Relay Driving—Figure 8.7 shows how a TTL output is connected to drive an external relay coil. The diode, D1, protects the internal circuit against the inductive 'kick-back' from the relay coil. Without the diode, the resulting high voltage spikes will damage the digital port. Note that the direction or polarity of the diode must be as shown in the diagram. Protection diodes must be able to respond very quickly and be able to safely absorb the coil's energy. Most standard 'switching' diodes fill these needs.

When large relays, contractors, solenoids or motors are involved, an additional driver or intermediate switching network can be employed.

Digital Isolation—When driving heavier loads than a TTL output is rated for, digital isolation modules can be used. These devices convert a standard TTL input via power transistors or triacs to switch high voltage/high current, AC or DC signals. Optical isolation provides high voltage separation between the load and the DA&C system, without using mechanical relays. Other modules are designed to monitor digital input signals while breaking the galvanic connection between the signal source and the measuring equipment. The modules not only isolate, but also convert the inputs to standard TTL levels that can be read by the DA&C system's digital input channels. Isolation is useful for safety, equipment protection and ground-loop interruption. Each module supports a single channel, allowing the flexibility to mix the various types when configuring a system. Special termination panels are available that accommodate either 8 or 16 isolated modules.

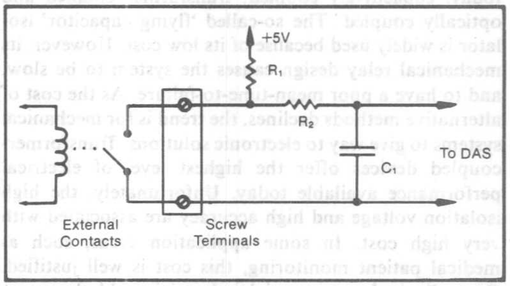


FIGURE 8.6. Contact Sensing and Wetting.

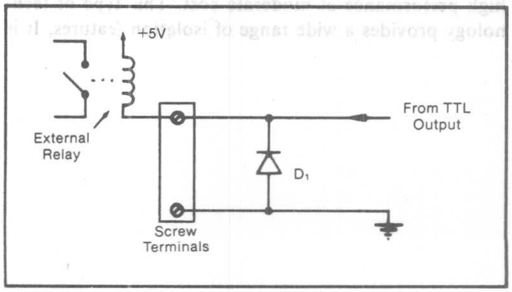


FIGURE 8.7. Relay Driving Circuit.

8

SIGNAL CONDITIONING ANALOG & DIGITAL

There are six basic optically isolated PCI modules: AC input, AC output, DC input, and DC output. The AC models are intended to monitor and control 120 and 240V power lines. Line voltage inputs can be directly connected to these devices. DC input models can accommodate almost any input level with the choice of an appropriate series resistor. The DC output devices are rated for up to 60V. Both the AC and DC models can switch loads up to 3A. More information on this family of opto-isolators is available by referring to the PCI-1100 series in the PCI-20000 section of this handbook.

Analog Isolation—Analog isolation as well as digital isolation can be used to protect people and equipment from contact with high voltages. Other applications include the breaking of ground loops or the removal of large common-mode signals. For example, if a thermocouple is connected to a motor armature, it could be at 240V above ground. However, the TC output voltage might be only 30mV. The 30mV (the actual signal) is a differential signal because it is applied to the + and − inputs of the data acquisition channel. On the other hand the 240V appears not as a differential signal, but, as a signal common to both + and − inputs. Common-mode voltages are referenced to the power supply ground. Standard analog input channels can only accept up to a 10V common-mode signal while remaining linear. Voltages above 30V are likely to damage the input components. An analog isolator separates a desired differential signal from an unwanted common-mode voltage.

Three major types of analog isolators are in wide use today: capacitively coupled, transformer coupled and optically coupled. The so-called 'flying capacitor' isolator is widely used because of its low cost. However, its mechanical relay design causes the system to be slow, and to have a poor mean-time-to-failure. As the cost of alternative methods declines, the trend is for mechanical systems to give way to electronic solutions. Transformer-coupled devices offer the highest level of electrical performance available today. Unfortunately, the high isolation voltage and high accuracy are associated with very high cost. In some application areas, such as medical patient monitoring, this cost is well justified. Generally, in the commercial/industrial world, the lower-cost optical isolators are more appropriate.

For use in general DA&C systems, optical isolators offer high performance at moderate cost. This type of technology provides a wide range of isolation features. It is desirable to keep all potentially high voltage signals outside the personal computer. For this reason, both digital and analog isolation should be provided on external termination panels. In addition to isolation, analog panels should also provide differential input gain stages, bridge-completion and excitation circuitry, as well as passive signal conditioning capabilities. Provisions for thermocouples, RTDs, strain gages and pressure transducers are also included in better systems.

More information on analog isolators can be found by referring to the PCI-20042T series in the PCI-20000 section of this handbook.

Motor Control—Many different types of motors are in common use today. When it comes to controlling these devices, specialized circuits are often required. Some applications, however, require only on-off operations. These can simply be driven by digital output ports, usually through optical isolators (load up to 3A) or with various types of contactors (relays).

In general, when variable speed is desired, either analog or digital outputs from the DA&C system are used to manipulate the motor through an external controller. A wide range of both AC and DC controllers are available from KB Electronics (Brooklyn, NY), and others.

Stepper-type motors are of particular interest in robotics, process control, instrumentation and manufacturing. They allow precise control of rotation, angular position, speed and direction. While several different types of stepping motors exist, the permanent-magnet design is perhaps the most common. The permanent magnets are attached to the rotor of the motor. Four separate windings are arranged around the stator. By pulsing DC current into the windings in a particular sequence, forces are generated to produce rotation. To continue rotation, current is switched to successive windings. When no coils are energized, the shaft is held in its last position by the magnets. In some applications these motors can be driven directly (via opto relays) by one of the DA&C system's digital output ports. The user would provide the required software to produce the desired pulses in proper sequence. The software burden could be reduced by driving the motor with a specially designed interface device. These units accept a few digital input lines representing the desired speed, rotation, direction and acceleration, etc. A full range of both motors and interfaces is available from companies such as Airpak Corporation (Cheshire, CT), Superior Electric (Bristol, CT) and others.

Section 9

APPLICATIONS, TOOLS & TECHNIQUES

INTRODUCTION

This section includes a collection of articles that address a number of important principles and techniques. The fundamentals presented earlier are now linked to available hardware and software products. In some cases it is appropriate to first have a basic knowledge of the specific hardware and or software items before studying these notes. When this is the case, detailed information can be found in the later, product sections of this handbook. A complete description, including specifications and system configuration information, for all PCI products will be found. Please refer to the Tab Index, Model Number Index or Subject Index to locate the desired product section.

TABLE OF CONTENTS

9

APPLICATIONS, TOOLS & TECHNIQUES
INTRODUCTION

ADVANCED APPLICATIONS FOR THE PCI-20007M-1 COUNTER/TIMER MODULE

FEATURING

- Terminal characteristics of the Intel 8254
- Alternate methods for frequency measurement
- Gate generation and speed measurement
- Enhanced, programmable, pulse generation
- Measuring the time between pulses

INTRODUCTION

The PCI-20007M-1 Counter/Timer Module finds wide application in fields such as interval timing, event counting, frequency measurement, speed monitoring and time-base or pulse (frequency) generation.

This module is based upon the Intel 8254 integrated circuit, which provides a spectrum of useful functions. To exploit many of the features of this module, however, can require a detailed knowledge and understanding of the "chip" itself. To assist users of the PCI-20000 system, an array of software tools has been created. This software provides a broad range of capabilities while eliminating the need for the user to program the 8254 directly.

Some of the characteristics of the 8254 merit clarification in order to avoid possible difficulties. For example: how does a counter load an initial value and record input pulses? This question relates to important features of the counter. This segment will offer useful background information and solutions to what some might call shortcomings of the 8254. Also included are alternate techniques for frequency measurement and external hardware approaches to speed monitoring, time measurements and programmable pulse generation.

TALKING TO THE COUNTERS

The PCI-20007M-1 consists of four independent counter channels and one "rate generator." Each counter has three input/output terminals: clock input (clk), gate input (gate) and output (out). Six different operating modes can be selected, each with unique characteristics. A complete description of the various modes can be found in the PCI-20007M-1 user manual. Of importance here is that the initial output level and its response to both clk and gate inputs are strongly dependent upon the mode chosen. Assumptions are often wrong. Please read the manual. In addition to the above I/O points, there are a number of on-board registers that permit software to set up the control parameters and to read the status of the counters. Each register function is explained in the user manual. The rate-generator consists of an 8MHz crystal oscillator driving two divide-by-"N" counters (separate from above), connected in series. The resulting output is a software-programmable pulse generator. By selecting the N values for the two counters, a wide range of pulse frequencies and duty-cycles can be obtained. All counters are 16-bit devices capable of representing numbers between 0 and 65, 535 (2^{16}).

In the context of the PCI-20007M-1, a pulse is defined as a digital signal which changes from a low-to-high-to-low state. An example of this is shown in Figure 1.

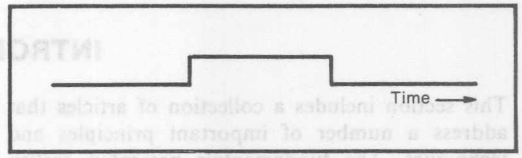


FIGURE 1. A Single Pulse.

Any of the counters on the PCI-20007M module can be configured to accumulate input pulses (input to clk). The Rate Generator can be used to produce an accurate time-base (input to gate). The number of input pulses counted will be directly related to the length of time that the gate input is held high.

The counter gates can be jumper programmed to respond to either signal or software inputs. In the signal mode, each independent counter can be inhibited from counting by holding its gate low (TTL 0). Internal pull-up resistors are not provided, so it is very important to drive the gates to the desired level. In the absence of an input, operation is not predictable. Remember, for the rate generator to be active, both associated counters must be gated "on."

COUNTING

Often it is desired to initialize a counter with a given number ("NUM" for example) and to either count inputs with reference to NUM or to generate a hardware output upon NUM inputs. The count value contained in the output register, decrements (reduces) with each clock pulse. However, the defined value, NUM, is not actually loaded into the counter until the first clock pulse occurs. The result is that the contents of the output register contains one less count than was actually applied. It is normally very easy to account for this condition, by adding 1 to the reading. But, what if fewer than two pulses are received? Under these conditions, the value contained in the register is unpredictable. Fortunately, there are solutions to this problem. One method using the PCI-20046S is outlined below.

Method 1:

a) Disable the counter............. (WRITE.CH with 0)
b) Set the Initial Count to 65, 535 ....(CNF.CNTR)
c) Start counting ....(WRITE.CH with 1)
d) Read the counter's STATUS & COUNT...(STAT.CNT)

If STATUS = 0, zero pulses have arrived
If STATUS = 1, read the counter (COUNT)
The number of input pulses = 65, 536 − COUNT

Sometimes it is required to count events in excess of 65,535. It may seem logical to cascade two counters to

produce a 32-bit device, but this is not quite as simple as might be expected. In the divide-by-N mode, the counter output is initially HIGH and it goes LOW when decremented to 1. The next input pulse causes the output to return to a HIGH state. Unfortunately, this is NOT the correct waveform to decrement the second counter. Intel defines an input pulse as a LOW to HIGH to LOW transition. As a result, 65K pulses will be missed before this condition is met. There is, however, an external way (Method 2, below) of correcting the hardware levels. In addition, a software solution is also available. This technique uses the fact that when a counter decrements to zero it automatically rolls-over to full scale (65, 535) and continues counting (method 3, below).

Method 2:

a) Use two counters.

b) Add a hardware inverter between the output of the first counter and the input of the second. This corrects the "sense" of the pulse.

c) Setup and read both counters following the general procedure suggested in Method 1. However, compute the total input pulses by applying the proper "weighting" factor to the second counter.

Method 3:

a) Use only one counter.

b) Follow the procedure in Method 1.

c) Read the counter often enough to detect a "zero crossing."

d) For each zero crossing add 65, 535 to the count computed in Method 1. That is: Total = (65,536 − COUNT) + (65,535 · number of zero crossings)

FREQUENCY MEASUREMENT

The structure of the PCI-20007M-1 is well suited to frequency measurement. Depending upon the application, several configurations are possible. The PCI-20046S Software Drivers support the most popular arrangement. This technique is suggested in Figure 2.

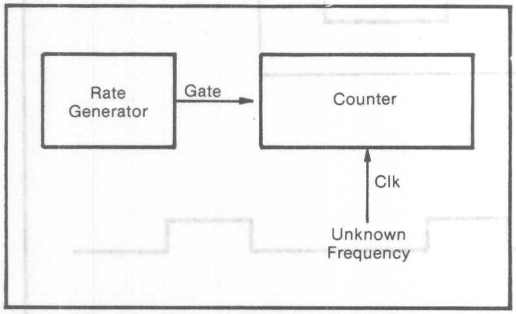


FIGURE 2. High Frequency Measurement Circuit.

This circuit counts input pulses for a known time interval (Gate time), as defined by the rate Generator. Gate times of 1ms to one second are very common. Clearly, the longer the measurement interval, the larger will be the accumulated count (for a given input frequency). Resolution and accuracy are proportionally enhanced. Maximum resolution is limited by the requirement that no more than 65K pulses be accumulated in any one measurement interval. The practical (useful) frequency range for this circuit extends from about 100Hz to beyond 8MHz.

An alternate approach is suggested in Figure 3. This circuit differs in that it counts high frequency clock pulses for the duration of the unknown input signal. The result is enhanced resolution at low frequencies. In the simple form shown here, the count value is a function of the input signal's duty-cycle (counting takes place while the input is high). Therefore, the exact value of the duty-cycle must be known to accurately compute the true input frequency. Given this information, practical measurements are possible from a small fraction of a Hertz to byond 1kHz. In fact the lower limit is only restricted by the time available to make the measurement (i.e., measuring 0.01Hz requires 100 seconds). Both of these circuits have an ultimate (maximum) resolution of one part in 65K. Note, however, that this resolution is not available at all frequencies within the measurement range. Programmers can implement this technique using the PCI-20046S, but it is not directly supported by the READ.FRQ command.

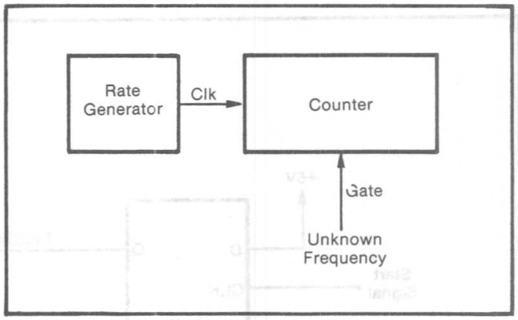


FIGURE 3. Low Frequency Measurement Circuit.

GATE GENERATION AND SPEED MEASUREMENT

As was stated above, Figure 3 requires a known duty-cycle. Very often this information is not available. For example, when measuring speed (i.e., motor rotation, conveyor travel, linear motion) with an optical or magnetic pickup, duty-cycle is variable. One solution is to use the circuit shown in Figure 4. This circuit uses flip-flops to convert the input signal into a positive pulse that is the width of a complete input cycle. Thus, measurements are now duty-cycle independent. An external logic input is applied to control the arming of the "gate generator." One of the PCI digital output lines can be used to produce this level internally. The selection of the Rate Generator frequency requires a balancing of two competing factors. The higher the frequency, the greater is the resolution of the final reading. However,

remember that the counter is limited to 65,535. "Speed" or "inputs per second" are proportional to:

$$\frac{\text{Frequency (Rate Generator)}}{\text{Number of Pulses Counted}}$$

WIDE RANGE, PROGRAMMABLE DUTY-CYCLE, PULSE GENERATOR

The Rate Generator within the PCI-20007M-1 can be programmed in both frequency and duty-cycle by selecting the "N" values for each of the two related counters. N_1 and N_2 designate the division factor for each counter. Because the crystal oscillator is 8MHz, the rate generator's output frequency is $8\text{MHz}/(N_1 \cdot N_2)$. Consider that the output waveform consists of a "low" and "high" level portion, represented by t_1 and t_2. Thus:

$t_1 = N_1 \cdot 125\text{ns}$, while $t_2 = N_1 \cdot 125\text{ns}\ (N_2 - 1)$.

Combinations of N_1 and N_2 allow a wide range of frequencies and duty-cycles to be selected. However, both parameters are interdependent.

Figure 5 shows a circuit configuration, using two counters in addition to the rate generator, that allows duty-cycle variation without interfering with the separately programmed frequency. Define the four "N" values:

N_1 = Rate Generator, counter 1
N_2 = Rate Generator, counter 2
N_3 = Independent counter 1 (operate in Mode 2)
N_4 = Independent counter 2 (operate in Mode 1)

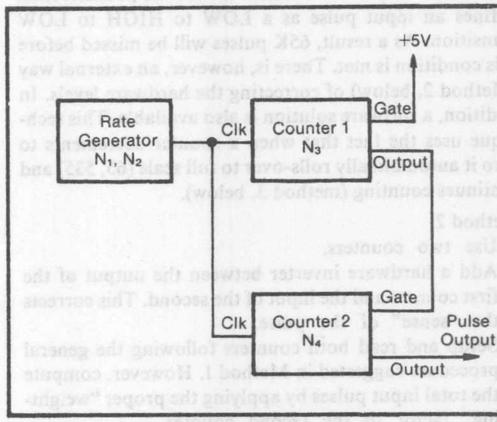


FIGURE 5. High Resolution, Variable Duty-Cycle Pulse Generator.

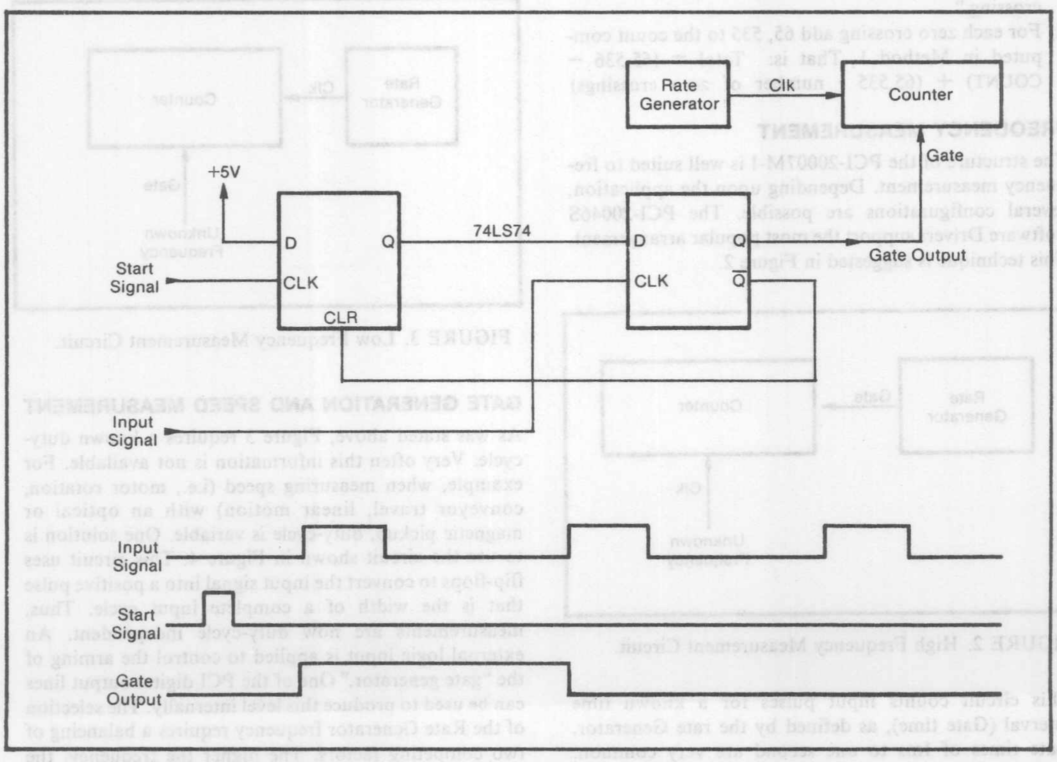


FIGURE 4. Gate Generator Circuit.

The frequency and duty-cycle for this circuit are:

Frequency = $8MHz/(N_1 \cdot N_2 \cdot N_3)$,

Duty-Cycle = $(N_3 - N_4)/(N_3 \cdot 100)$, in percent.

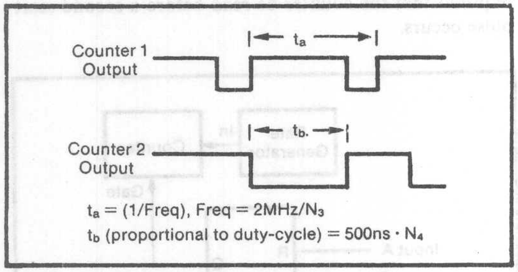


FIGURE 6. Waveforms for Variable Duty-Cycle Generator.

MEASURING THE TIME BETWEEN PULSES

The following applications are related to the measurement of time *between* pulses. Three different circuits are shown:

- Low speed, time between two pulses on the same line.
- High speed, time between two pulses on the same line.
- Time between two pulses on different lines.

The trick is to generate an appropriate gate signal corresponding to the time between two input pulses. This is suggested in Figure 7.

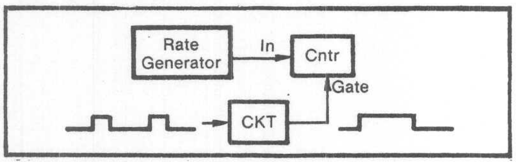


FIGURE 7. Basic Counting Circuit.

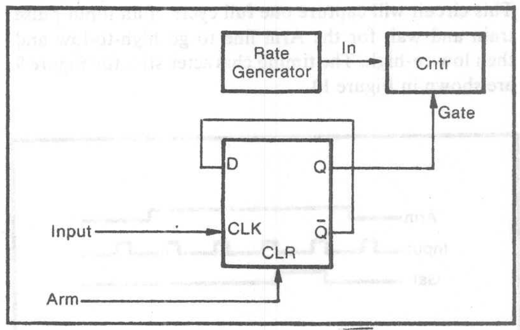


FIGURE 8. Simple Time Interval Measurement Circuit.

Before choosing the frequency of the rate generator, the user must first know the maximum duration of time that the pulse at the gate will exist. The product of the maximum gate time and the frequency must not exceed the 16-bit counters' limit of 65,535. However, the higher the rate generator frequency, the greater the resolution of the interval measurement will be.

Figure 8 shows a simple circuit for converting successive input pulses, on a single line, to a gate signal. This circuit is slow because once armed, the counter must be read before a third input pulse occurs. If this condition is not satisfied, the counter will be restarted and an incorrect count will result. Another alternative is to bring the Arm signal low after the second input pulse. This condition can be detected by reading the counter output until a stable count is observed.

Figure 9 shows a variation on the above circuit. Here, the acquisition of the desired time interval is latched with hardware. Thus, the measurement of the desired data is assured without intervention from the host computer.

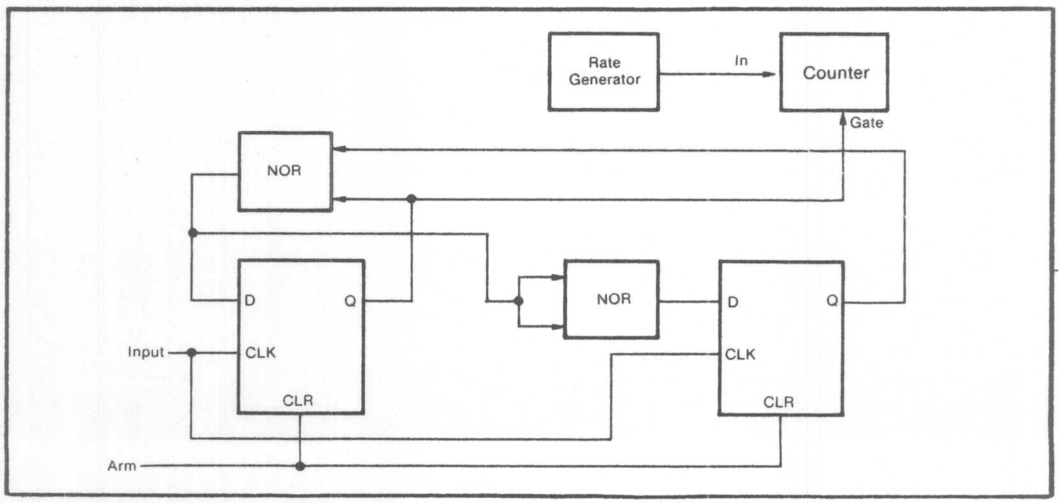


FIGURE 9. Fast Time Interval Measurement Circuit.

This circuit will capture one full cycle of an input pulse train and wait for the Arm line to go high-to-low and then low-to-high. The timing characteristics for Figure 9 are shown in Figure 10.

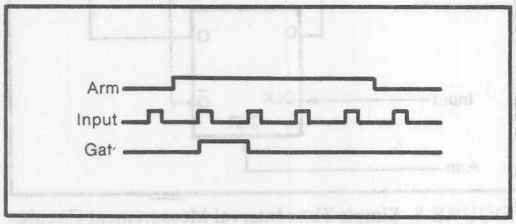


FIGURE 10. Timing Characteristics of the Circuit in Figure 9.

In both of the above circuits, the time interval was measured between two pulses occurring on the same input line. Figure 11 shows a circuit that performs the same function with separate START and STOP inputs.

Input A gives the START pulse, and Input B gives the STOP pulse. The time between the START/STOP signals is what this circuit is measuring. As in Figure 3, this circuit is most useful for low-speed pulses. It is again required that the counter be read before a second start pulse occurs.

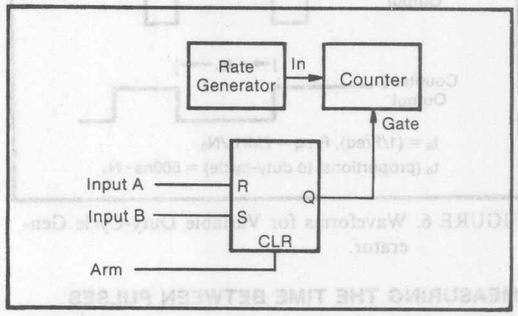


FIGURE 11. Double-Input Time Interval Measurement Circuit.

DATA ACQUISITION, TEST, MEASUREMENT, ANALYSIS AND CONTROL USING PCI-20040S-1 "LABTECH NOTEBOOK"

INTRODUCTION

PCI-20040S-1 is the model number for the industry standard LABTECH Notebook, designed for PCI products.

LABTECH Notebook is an integrated, general-purpose software package for data acquisition, monitoring, and real-time control (see Figure l). It runs on the IBM® PC, XT, AT and other PC-compatible computers.

Notebook couples to many of the components within the PCI-20000 system including the PCI-20001C Series Carriers and the PCI-20002M, PCI-20003M, PCI-20004M, PCI-20005M, PCI-20007M, PCI-20019M, and PCI-20021M Series of Modules. Parts of the PCI-3000 Family are also compatible. These include the PCI-3002 and PCI-3003 Master Enclosures and the PCI-3307-I and PCI-3307-2 I/O boards.

DATA ACQUISITION AND ANALYSIS USING LABTECH NOTEBOOK

The following example will illustrate the ease with which Notebook can be used to perform data acquisition and analysis tasks.

Imagine that we are performing a series of tests involving a gas contained in a sealed vessel. During the test, a chemical reaction occurs, causing both the pressure and temperature in the vessel to rise. The pressure is measured using a sensor located inside the sealed vessel. At the same time, the temperature both inside and outside the vessel is monitored using thermocouples.

Given this experimental setup, we will use LABTECH Notebook to perform the following tasks:

- First, we will perform a data acquisition run which will last for a period of three seconds, during which data from the sensors will be sampled at a rate of 100Hz.
- Once the pressure data has been taken, its time-derivative will be computed and the maxima of both the pressure and pressure-derivative curves will be found.
- In addition, we will fit the data to a mathematical model of the gas-heating process.
- After the processing is complete, we will create a database so that we may catalogue the results of this and subsequent experiments.

SETTING UP AND PERFORMING THE DATA ACQUISITION

Once the data acquisition problem has been defined, we can begin setting up Notebook. First we select the SETUP function from the main menu.

Our setup will be divided into three parts, represented by the first three choices in the SETUP menu: CHANNELS, FILES and DISPLAY. The real-time capabilities of Notebook are divided among these three functions. CHANNELS defines the transfer of data between the PCI hardware and Notebook, FILES is used to setup the

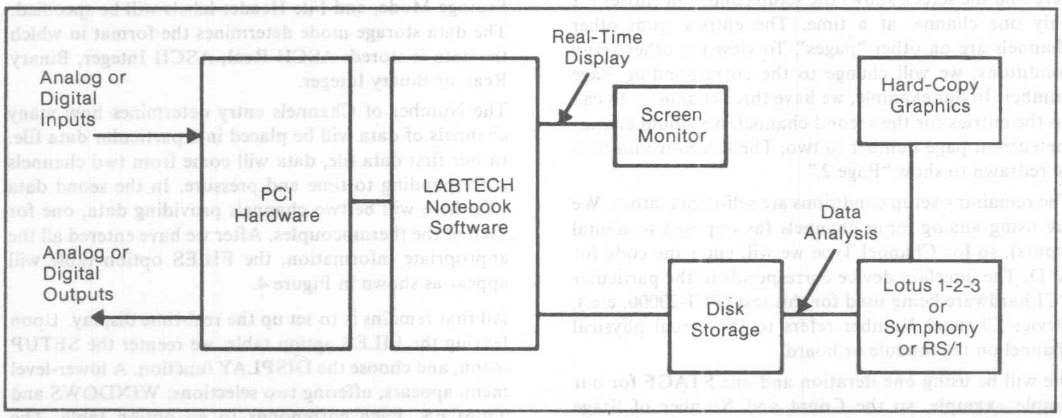


FIGURE 1. A Functional Diagram of LABTECH Notebook.

transfer of data between Notebook and the disk storage files, and DISPLAY controls Notebook's real-time display.

We will begin by selecting CHANNELS from the SETUP menu. A lower-level menu appears, offering two choices: NORMAL or HIGH-SPEED. Since for our example, data will be taken at a rate of 100Hz, the NORMAL mode will be adequate. Having selected NORMAL, an option table appears on the screen (see Figure 2).

```
Current Value: Analog Input
              NORMAL DATA ACQUISITION / CONTROL SETUP
Number of Channels [1..100]                      10
Current Channel [1..10]                           1
Channel Type                           Analog Input
Channel Name
Interface Device                      0: Demo Board
Interface Channel Number [0..15]                  1

Input Units                                DC Volts
Input Range                                     0-5
Scale Factor                                  1.000
Offset Constant                               0.000
Buffer Size                                    2048
Number of Iterations                              1
Number of Stages [1..4]                           1

Sampling Rate, Hz                            10.000
Stage Duration, sec. [0.0..1.0E+09]          29.500
Start/Stop Method                            Normal
Trigger Channel                                   0
Trigger Pattern to AND [0..255]                   0
Trigger Pattern to XOR [0..255]                   0
```

FIGURE 2. The CHANNELS Option Table.

By examining the current values for each of the setup conditions we can determine which need to be changed to conform to our example problem.

The first entry is the Number of Channels. Since we will be taking data from three sources (a pressure sensor and two thermocouples), we will use three of the PCI's channels. In addition, we will use LABTECH's time stamping capability in a fourth channel. We indicate this by entering a 4 in the table.

The next entry is the current channel. We should note here that the screen shows the setup condition entries for only one channel at a time. The entries from other channels are on other "pages". To view the other setup conditions, we will change to the corresponding Page number. In our example, we have three channels. To call up the entries for the second channel, we would change the current page number to two. The screen would then be redrawn to show "Page 2."

The remaining setup conditions are self-explanatory. We are using analog input channels (as opposed to digital inputs), so for Channel Type we will enter the code for A/D. The interface device corresponds to the particular PCI hardware being used for this test (PCI-20000, etc.). Device Channel Number refers to the actual physical channel on the module or board.

We will be using one iteration and one STAGE for our simple example, so the Count and Number of Stage entries will be set to one. The Scale Factor and Offset Constant will be set to convert the voltage output from the sensors to engineering units such as PSI or degrees. As noted above, tthe Sampling Rate will be 100Hz and the Run Duration will be three seconds.

Let us suppose that our experimental setup involves a switch which initiates the chemical reaction in the sealed vessel, and that the switch is also connected to a digital input of the PCI system. We can use this switch as a trigger for starting the data acquisition. For Starting Method, we choose a Triggered start instead of a Normal or Time-delayed start. The CHANNELS setup is now complete. The resulting option table is shown in Figure 3.

```
Current Value: Analog Input
              NORMAL DATA ACQUISITION / CONTROL SETUP
Number of Channels [1..100]                       4
Current Channel [1..10]                           1
Channel Type                           Analog Input
Channel Name                               Pressure
Interface Device                       2: PCI-20000
Interface Channel Number [0..15]                  1

Input Units                                DC Volts
Input Range                                    0-10
Scale Factor                                  4.500
Offset Constant                              12.000
Buffer Size                                    2048
Number of Iterations                              1
Number of Stages [1..4]                           1

Sampling Rate, Hz                           100.000
Stage Duration, sec. [0.0..1.0E+09]           3.000
Start/Stop Method                           Trig On
Trigger Channel                                   0
Trigger Pattern to AND [0..255]                   1
Trigger Pattern to XOR [0..255]                   0
```

FIGURE 3. The CHANNELS Option Table After Setup.

When we leave the option table, the SETUP menu reappears. This time we choose FILES from the menu, and another option table appears.

This option table contains setup conditions which will control the storage of our data on a disk. Again, we enter the values appropriate for the example problem. We will set up one data file for the pressure data, and another data file for both sets of temperature data. To do this, we will enter 2 as the Number of Data Files.

The FILES option table has a "page" system similar to that of the CHANNELS. Here, each page refers to a separate data file. For each data file a File Name, Data Storage Mode, and File Header labels will be specified. The data storage mode determines the format in which the data is stored: ASCII Real, ASCII Integer, Binary Real, or Binary Integer.

The Number of Channels entry determines how many channels of data will be placed in a particular data file. In our first data file, data will come from two channels corresponding to time and pressure. In the seond data file, there will be two channels providing data, one for each of the thermocouples. After we have entered all the appropriate information, the FILES option table will appear as shown in Figure 4.

All that remains is to set up the real-time display. Upon leaving the FILES option table, we reenter the SETUP menu, and choose the DISPLAY function. A lower-level menu appears, offering two selections: WINDOWS and TRACES. Each correponds to an option table. The WINDOWS option table determines the size, number, and color of the graphs to be displayed on the screen during the data acquisition run.

```
Current Value: ASCII Real
                         FILES SETUP

Number of Data Files [0..12]                      2
Current Data File [1..2]                          1
Data File Name                                Pres3
Data Storage Mode                        ASCII Real

Number of Header Lines [0..4]                     4
Header Line 1                         Pressure Data
Header Line 2                               Test #3
Header Line 3                                 $DATE
Header Line 4                                Oxygen
Number of Channels in File [0..35]                2
Channel  Number [1..2]                            4        1
Channel Name                                   Time Pressure
Channel Units                                  Sec.     Psig
Field Width (ASCII Files)                        10       10
Decimal Places (ASCII Real Files)                 2        3
```

FIGURE 4. The FILES Option Table After Setup.

TRACES is used to set up the data traces which will be drawn in the windows. We will use WINDOWS first, and set up two windows: one large one, in which we will display the pressure data, and a smaller one, in which data from the two temperature sensors will be displayed.

Once again, we enter an option table. For the first entry, Number of Windows, we enter a 2. The table then breaks out into two columns, one for each window.

We choose position limits for each window. All windows are in the shape of rectangles. The limits determine where the edges of the rectangles will appear on the screen during the data acquisition run. The limits must be in the range from 0 to 1, where 0 represents the left edge of the screen (for the left and right limits) or the bottom edge of the screen (for the lower and upper limits), and 1 represents the right edge or upper edge of the screen.

We will also enter the axis scale ranges in this option table, as well as labels for both axes. In addition, the color of each window will be chosen. Up to eight colors are available on systems with color monitors.

The WINDOWS setup is now complete; Figure 5 shows the option table. We may leave this option table and enter the TRACES option table.

```
Current Value: 2
                         WINDOW SETUP
Number of Windows [0..15]                    2

         Window Number                       1         2
Left  Limit, x0 [0.0..1.0]               0.100     0.100
Lower Limit, y0 [0.0..1.0]               0.100     0.750
Right Limit, x1 [0.0..1.0]               0.990     0.990
Upper Limit, x0 [0.0..1.0]               0.600     0.990
Y Axis Title                          Pressure      Temp
X Axis Title                              Time      Time
Length of Time (X) Axis in sec.          3.000     3.000
X Tic Start Value                        0.000     0.000
X Tic End Value                          3.000     3.000
Number of X Tics [0..11]                     4         4
Y Tic Start Value                        1.000     1.000
Y Tic End Value                          4.000     4.000
Number of Y Tics [0..11]                     4         4
Window Color                              Blue    Yellow
Scroll Size [0.0..1.0]                   1.000     1.000
```

FIGURE 5. The WINDOWS Option Table After SETUP.

Although we have set up only two windows, the Number of Traces for this run is three, since we wish to display the data from the first three channels. In the option table, the Channel Number for each trace will be specified. In addition, we will enter a window number for each trace.

Since we want the pressure trace to appear in the larger, lower window, we enter Window Number 1. The number 1 corresponds to the left-most column of the WINDOWS option table. The two temperature traces will be places in Window Number 2. We will also choose a color for each trace. When this is done, the TRACES option table will appear as shown in Figure 6, and we can return to the SETUP menu.

```
Current Value: 3
                         TRACE SETUP

Number of Traces [0..50]                     3
         Trace Number                        1         2         3
Window Number [1..5]                         1         2         2

Line Color                               Green       Red     Black
Line Type                                Solid      Bold     Solid
Data Point Symbol                         None      None      None

Y Channel Number [1..5]                      1         2         3
Y Minimum Displayed Value                0.000     0.000     0.000
Y Maximum Displayed Value               10.000    10.000    10.000

Trace Type                              T vs Y    T vs Y    T vs Y
For Meters Only:
  Number of Decimal Places                   2         2         3
For X vs Y Only:
  X Channel Number [1:5]                     1         1         1
  X Minimum Displayed Value              0.000     0.000     0.000
  X Maximum Displayed Value              4.000     4.000     4.000
```

FIGURE 6. The TRACES Option Table After Setup.

Now that the entire setup has been completed, we can use VERIFY to perform a global checking routine to insure that setup conditions are consistent between the three SETUP areas. VERIFY displays a summary of the setup condition, and lists inconsistencies between the option tables. For example, if we had asked for a trace from Board 2, Channel 3 in the TRACES option table, but had not entered this channel in the CHANNELS option table, VERIFY would display an error message.

SAVE/RECALL can be used to save the current setup conditions (that is, the contents of the setup files created and used internally by Notebook), or to RECALL a set of setup conditions that has been previously saved.

STARTING THE RUN

LABTECH Notebook is now ready to perform the data acquisition we specified in the problem statement above. To begin the run, we return to the main menu and select

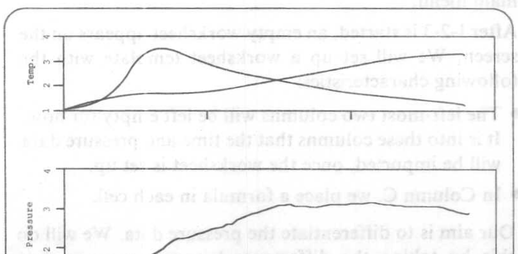


FIGURE 7. The REAL-TIME Display.

the GO function. The screen will clear and two windows will appear on the screen.

Notebook now waits for the trigger we specified in the CHANNELS option table. Once the switch has been thrown, the gas in the sealed vessel will begin to heat up and Notebook will begin to take data. If we had not specified a triggered start, Notebook would have begun to take data after GO was invoked. As the data is taken, it is displayed in the windows on the screen. At the end of the run the data is written to a data file on the specified disk drive. The resulting display is shown in Figure 7.

DATA MANIPULATION AND GRAPHING

We many now begin analyzing the pressure data we have taken. This can be done in several ways, including adding calculated channels to do the data reduction while the data is being collected. For this example, we will use the Lotus 1-2-3 spreadsheet program. Using 1-2-3 we will set up a single worksheet to perform all our analysis tasks.

The worksheet is a matrix of "cells." Columns are denoted by letters, and rows are denoted by numbers. We may enter numbers, characters, or formulas in the cells using the same method we learned for the SETUP option tables. We move the highlighted rectange to the appropriate cell using the cursor control keys, then enter the value by typing it and pressing the ENTER key.

The formulas contain variables which are cell addresses (e.g., A4, D23). The displayed value of a cell containing a formula will be the result of that formula when the variables are replaced by the current values of the addressed cells. When the values of these addressed cells change, the displayed value of the formula cell will change accordingly.

A particularly powerful feature is 1-2-3's use of worksheet "templates." A worksheet format can be standardized so that each time it is used, formulas, labels, and graph features will be entered and arranged in the same way.

We will take advantage of this feautre in using 1-2-3 to solve our data analysis problem. With the Lotus diskette in place, the 1-2-3 program is invoked directly from Notebook by selecting the ANALYZE function from the main menu.

After 1-2-3 is started, an empty worksheet appears on the screen. We will set up a worksheet template with the following characteristics:

- The left-most two columns will be left empty for now. It is into these columns that the time and pressure data will be imported, once the worksheet is set up.
- In Column C, we place a formula in each cell.

Our aim is to differentiate the pressure data. We will do this by taking the differences between pressure data points and dividing them by the corresponding time differences. If we did this on a point-by-point basis, however, the result would be a very noisy derivative curve. In the cells of this column, therefore, we will enter a formula to perform a five-point smoothing function. The pressure data will be imported into Column B. To do this, we only need to type a formula into one cell. The 1-2-3 "Copy" command is then used to copy it into each of the other cells in the column.

- In Column D, we will place the appropriate formula in each cell to compute the derivative using the "smoothed" data of Column C.
- Two more values need to be found, the maximum pressure and the maximum pressure derivative value. We will compute these maxima by placing the 1-2-3 "Max" command (with appropriate arguments) in two separate cells in Column E.

Once the template has been set up, we can import our data file into the worksheet. As the data is placed in Column B, the formulas of Columns C, D and E are re-calculated. Column D will now contain the calculated time-derivative of the pressure data. The two cells in Column E will contain the maximum values for the pressure and pressure derivative data. The filled-in worksheet is shown in Figure 8. These calculated data, along with the rest of the worksheet, can be saved for later use.

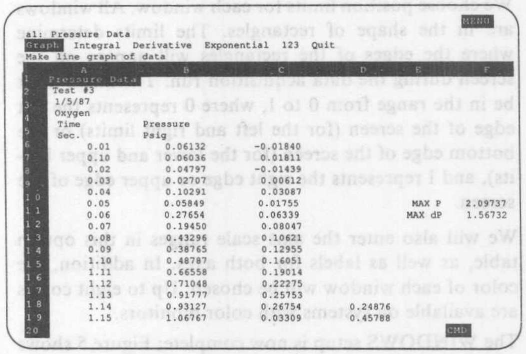


FIGURE 8. The Worksheet After a Data File Has Been Imported.

Note that all of these functions—five-point smoothing, differentiation, finding the maxima of the pressure and its derivative—could all have been done with LABTECH using calculated channels (in real-time if desired).

GRAPHING WITH 1-2-3

We can also use 1-2-3 to create a graph of the experimental and calculated data. The time data from Column A will be used for the X axis, and the pressure and pressure derivative data will be plotted against time. 1-2-3 allows the use of different colors or intensities to denote different categories of data. Titles and appropriate labels may also be included. The graph setup conditions can be saved along with the worksheet template.

After this graph has been set up exactly the way we want it, 1-2-3's PrintGraph function may be used to produce a publication quality hard-copy plot of the test results. This hard-copy graph may be plotted on either a dot-

matrix printer or a pen plotter. Figure 9 shows the final results, obtained with a multi-color pen plotter.

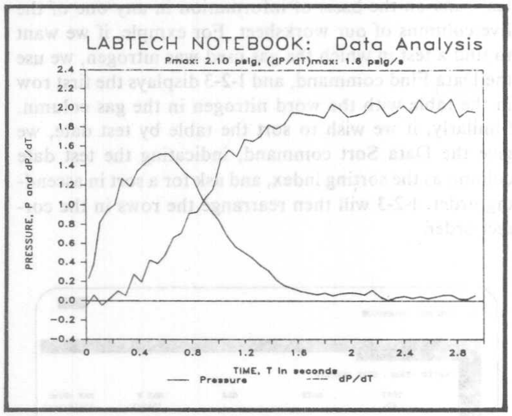


FIGURE 9. The Lotus 1-2-3 Graph.

Our data manipulation and graphing using 1-2-3 are now complete. Note that only one column of data has been brought in from the LABTECH Notebook data file. The derivative and maxima data are computed by 1-2-3 from the raw data.

USING LABTECH NOTEBOOK'S CURVE-FITTING FEATURE

Notebook's nonlinear regression function can be used to fit our experimental pressure data to a predetermined mathematical model. For example, suppose we had developed the following model for the increase in gas pressure due to heating:

$$P = RT/V\left(3 - \frac{2/a}{(t+1)\exp(b\,t)}\right)$$

where: P = Pressure
R = Gas Constant
T = Temperature
V = Volume
t = Time
a,b = Parameters to be determined

We will use LABTECH Notebook to find reasonable values for the parameters, a and b. To invoke the nonlinear regression function, we select CURVE-FIT from Notebook's main menu. A lower level menu appears, offering two choices: EXECUTE and SETUP. SETUP will be selected first. An option table similar to those we used for the data acquisition setup appears, as shown in Figure 10.

```
Current Value: x0 := 3.0-2.0/(p1*x1*exp(p2*x1)+1.0)
                         CURVEFIT SETUP
Model: x0 := 3.0-2.0/(p1*x1*exp(p2*x1)+1.0)

Number of data points [3..100]               100
Number of data variables [2..11]               2
Data file name                           a:exp10
Maximum SSQ evaluations [1..32767]           500
Iterations between printing [1..32767]        20
Do error analysis                             No
Theory curve #1 data values                    1
Theory curve #2 data values
Theory curve #3 data values
Number of parameters in model [2..10]          2
Parameter 1: start, tolerance              0.500       0.100
Parameter 2: start, tolerance              6.000       0.100
```

FIGURE 10. The CURVE-FIT Option Table.

Our theoretical model belongs in the first entry. The entire equation is entered, substituting the variable names with the labels X0 through X9, and the parameter names with the labels P1 through P9.

Next, we will enter the number of data points to be used by the curve-fitting routine. During our data acquisition run, we took 100 data points-per-second, for three seconds. Since we intend to use all of these, we will enter 300 here. If we entered the number 200, the first 200 points in the data file would be used, and the last 100 points would be ignored. In addition, the data file name, number of variables, and number of parameters will be entered. There are two variables and two parameters in our model.

For each parameter, a starting value and tolerance will be chosen. CURVE-FIT uses the starting values as initial estimates in its regression routine. The routine will continue until the change in the parameter between iterations is less than the tolerance value. Of course, if the model is not appropriate for the data, the routine may never be able to find appropriate parameter values. For this reason, we will enter a Maximum SSQ (Sum of Squares) Evaluations value. This value determines the maximum number of times CURVE-FIT will repeat the regression routine.

We can also choose to have an error analysis performed on the quality of the fit to the model. The output from this analysis consists of estimates of the parameter standard deviation and a parameter correlation matrix. The Do Error Analysis entry will be set to 1 if we wish to have the routine perform this analysis. The CURVE-FIT setup is now complete. After leaving the option table, we select EXECUTE from the lower-level menu. The routine will begin, and the screen will display parameter information as the routine progresses. CURVE-FIT creates five output files, containing:

- Final values of the parameters, along with estimated standard deviations.
- Experimental data points used by the routine (one column for each variable).
- Theoretical data computed from the model using the final parameter values.
- Final residual values for each of the experimental data points.
- A report containing the results of both the non-linear regression routine and the error analysis. This file is a readable summary of the regression procedure and results.

The screen output prompts us when CURVE-FIT is finished, and displays the final parameter values, as shown in Figure 11.

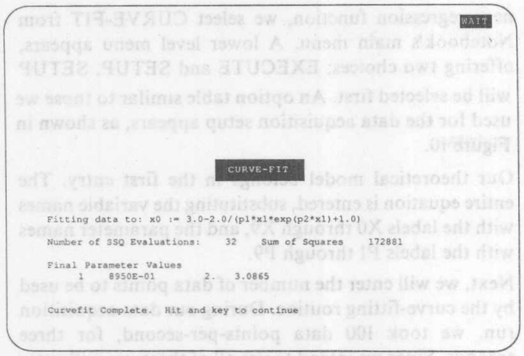


FIGURE 11. The CURVE-FIT Screen Display.

We can use Lotus 1-2-3 to create a graph comprising the theoretical and experimental data for our experiment. LABTECH Notebook has an "autostart" template (a worksheet template which is automatically started when the 1-2-3 program is invoked from the main menu) which can be used for this purpose. Figure 12 shows the graph created by this worksheet template using the experimental and theoretical data from our example.

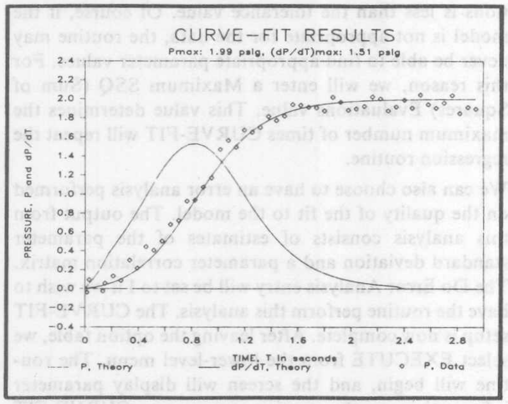


FIGURE 12. 1-2-3 Graph Using CURVE-FIT Output.

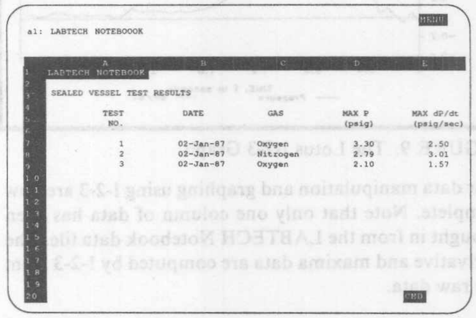
CREATING A DATA BASE OF TEST RESULTS

In a typical laboratory or testing situation, the example above would be only one of many performed in the same testing environment. Unless some means of organizing the ouptut files can be found, the resulting quantity of data files and analysis reports quickly becomes unwieldy.

1-2-3 facilitates the creation of a test result data base. A separate worksheet, such as the one in Figure 13, can be used to store important information for identifying a particular test run.

In the figure, test runs are identified by test number, test data, type of gas used and maximum pressure derivative value.

If the worksheet were to contain a hundred such rows of information, it would be difficult to visually search the table for information about a particular test. In this case, the 1-2-3 Data commands can be used to locate specific test runs on the basis of information in any one of the five columns of our worksheet. For exmple, if we want to find a test in which the gas used was nitrogen, we use the Data Find command, and 1-2-3 displays the first row in the table with the word nitrogen in the gas column. Similarly, if we wish to sort the table by test date, we give the Data Sort command, indicating the test date column as the sorting index, and ask for a sort in ascending order. 1-2-3 will then rearrange the rows in the correct order.



TEST NO.	DATE	GAS	MAX P (psig)	MAX dP/dt (psig/sec)
1	02-Jan-87	Oxygen	3.30	2.50
2	02-Jan-87	Nitrogen	2.79	3.01
3	02-Jan-87	Oxygen	2.10	1.57

FIGURE 13. A Data Base Worksheet.

PRODUCTION LINE TESTING WITH LABTECH NOTEBOOK

Once the setup conditions and data analysis worksheets have been developed, LABTECH Notebook can be configured for production-line testing very easily. All that is required is a very short program written in Notebook's programming option language, MAGIC/L™. This program can either be entered at the console and executed immediately, or stored in an autostart file which will be executed when the computer's power is turned on.

A program to collect an entire shift's worth of data, say 100 pressure vessel tests, might look like the following:

```
 iter 100   ;Repeat routine 100 times
     dogo   ;Invoke GO function
    r_123   ;Invoke 1-2-3, perform analysis
     loop   ;Return to first command
```

This simple program will perform 100 data acquisition runs. Each data acquisition will be automatically triggered by an electrical signal from the production line. After each run, the data files for the run will be immediately analyzed. The results from all the runs will be accumulated in a data base file (using 1-2-3's macro capability), thus completely automating the shift's testing.

CAPTURING AND ANALYZING TRANSIENT WAVEFORMS WITH A PERSONAL COMPUTER

Introduction

Until recently, the acquisition of transient analog data in the 100Hz to 25kHz region was dominated by storage oscilloscopes and expensive, dedicated, data acquisition systems (DAS). Any non-repetitive phenomenon is a candidate for analysis with this type of system. Some examples include:

- Mechanical shock and vibration studies
- Monitoring the onset of high speed chemical reactions such as explosions.
- Measuring medical signals such as QRS complexes, HIS bundle response, and nerve impulses
- Analyzing audio transients

In addition to capturing transient signals, some processing oscilloscopes and dedicated DAS's also provide a limited level of analysis capability. Usually, however, a computer is required to archive the data, to do special purpose analysis and to generate graphs and reports. These types of systems are usually linked to a host computer via an RS-232 or IEEE-488 bus. As a result, speed is severely limited. Obviously, it would be far more efficient to have the computer as a part of the system that took the data in the first place. The recent availability of powerful, low cost, personal computers (PCs) has enhanced the above methods, but more importantly has opened the door to new techniques.

The PCI-20000 System

State-of-the-art data acquisition systems are now available as board level products. For example, the PCI-20000 can plug directly inside any IBM-compatible PC, forming the basis of a complete data acquisition, test, measurement and control system. With appropriate software, a package can be put together to acquire, display, measure and analyze transient signals for a fraction of the cost of yeaterday's scopes or DASs. Besides the advantage of lower cost, the PCI-20000 approach has greater overall speed and versatility. The system's data acquisition rate can be as high as 89kHz in the PC or PC/AT.

The PCI system is based upon a motherboard, or Carrier approach. The carrier plugs into the computer and modules plug onto the carrier adding desired functionality. A large family of modules is available. Up to three Instrument Modules can plug into a given carrier. Typically, each module performs one data acquisition function such as analog input, analog output, counter/timer, digital input/output, etc. Multiple carriers can be used if required to accommodate a very large application. Thus, the PCI-20000 combines the high speed of a computer-based product with the flexibility, modularity, and expandability of a plug-in board device.

The Transient Recorder

This describes how to combine the IBM PC (or compatible with the PCI-20000 to produce a low cost, high performance transient data recorder. A complete program listing is included. The features of the transient capture system include:

- Menu driven, no programming required
- 12-bit resolution and linearity
- Sample rates from 150Hz to 50kHz
- Hardware slope and level triggering
- Capture of up to 25,000 samples
- Selection of one of eight channels
- Auto-scaled graphic display
- Data stored and recalled to and from disk

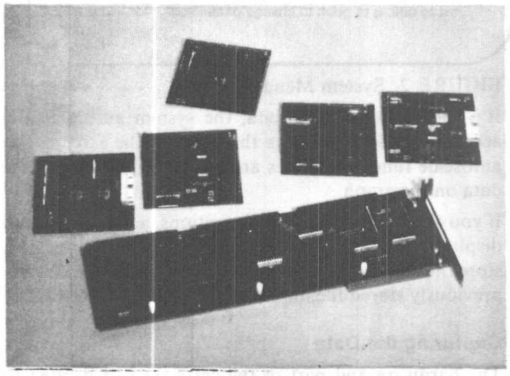


FIGURE 1. The PCI-20000 System.

The hardware (PC and PCI) is combined with a BASIC language program to provide the fundamental capabilities that a transient capture system needs. The system can acquire data, display data, store data to disk, and recall stored data from disk. The data is stored in ASCII format so that it can be manipulated, analyzed and printed by other BASIC programs, or word processors. There is an extensive library of PCI software drivers to interface with the PCI hardware. These drivers make it easy for a programmer to "talk" to the hardware, without being familiar with the details of multiplexers, programmable-gain amplifiers, sample holds, analog-to-digital converters, etc. The program shown here makes wide use of this library.

Software Highlights

The program begins by drawing a box for the graph on the screen, and by displaying the default data acqusition parameters along with the system menu (Fig. 2). The data acquisition parameters include:

- sample rate
- trigger level and slope
- channel number
- amount of data to be taken

9 APPLICATIONS, TOOLS & TECHNIQUES CAPTURING TRANSIENT WAVEFORMS

The menu allows you to:

- exit the program
- to set the acquisition parameters
- to acquire and display a data set with the above parameters

to execute one of the file operations

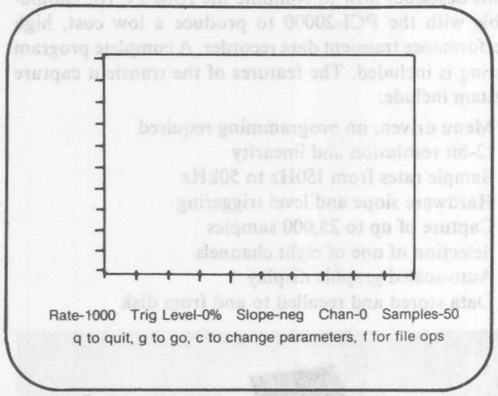


FIGURE 2. System Menu.

If you choose to take data, the system automatically acquires, scales and plots the data on the screen. The autoscale function insures an optimum fit for all of the data on the graph.

If you choose one of the file operations, a second menu is displayed. You can choose to look at the disk directory, store the current data set to a file, recall a data set from a previously stored file, or re-plot the current data set.

Capturing the Data

The hardware and part of the software is dedicated to the task of capturing the transient data. In this type of system, the amount of software which actually interacts with the hardware is small, yielding maximum speed. The majority of the work related to capturing the data is done directly by the hardware. Most of the software is involved in generating the data display and in supporting the menu-driven user interface.

The fundamental requirements of a useful transient capture system are:

1) **The Ability To Begin Data Capture At The Right Time**—Because the PC has finite memory space, acquiring data at high speed can quickly fill up available memory. Therefore, it is important to be able to "trigger" the beginning of data capture at the point of interest.

2) **The Establishment Of A Stable Data Acquisition Rate**—Data is usually time correlated. Many algorithms, such as the FFT, require the exact interval between data points to be known.

In this system, the capture of data begins when the input signal crosses a specified threshold in a given direction (as with an oscilloscope). Once this trigger condition has been met, the acquisition is paced by a crystal-controlled rate generator. Thus, both of the above criteria are met; the slope/level triggering insures that the event of interest is captured, and the crystal-controlled time base insures a stable, jitter-free acquisition rate.

The triggering, timing and data acquisition functions are all available, in hardware, with PCI-20000 Instrument Modules. The hardware is configured by selecting the appropriate modules from the wide variety available, and by plugging them into a carrier. For this high speed system the choices are:

- PCI-20001C-1 Carrier for the IBM PC
- PCI-20007M-1 Counter/Timer/Rate Generator Module

— or —

- PCI-20041C-2 High Performance Carrier for the IBM PC

—and—

- PCI-20019M-1 High Speed Data Acquisition Module
- PCI-20020M-1 Trigger/Alarm Module.

Note that if you use the PCI-20041C-2 Carrier, no PCI-20007M Rate Generator module is required. This function is built into the PCI-20041C-2.

To facilitate connecting the input signals to the modules, termination panels and cables are also selected:

- PCI-20057T-1 High Density Analog Termination Panel.
- PCI-20012A-1 6-ft Shielded Analog Cables, 2 required.

Finally, the PCI Language Software Support Package mentioned earlier, makes implementing the desired application much easier. While BASIC is used in this example, Turbo-Pascal, C, and ASYST are also available.

- PCI-20046S-1 Basic Language Modular Software Support Package.

Appendix A contains the details on how to configure the individual modules and plug them together. Figure 3 shows a block diagram of the system.

Hardware Description

The carrier contains space for up to three plug-in modules. It also has facilities which allow one module to pass both data and synchronization signals to other modules on the carrier. These unique abilities are made possible by the Intelligent Instrumentation Interface Bus (I^3 Bus, patent pending). Each module has a SYNCIN and a SYNCOUT connection. Jumpers on the carrier determine the source of a module's SYNCIN, and the destination of its SYNCOUT. The ability to pass hardware timing signals from one module to another results in a very close timing relationship between the various modules. This is a key factor in the performance of this system.

As seen below in Figure 3, the Rate Generator's output connects through the Trigger/Alarm Module (via the I^3 Bus) to become the timing source for the Analog Data Acquisition Module (A/D). When the desired input conditions are satisfied, the Trigger/Alarm Module 'gates' the Rate Generator signal to the A/D. Once 'triggered', the A/D continues to convert the input data at the rate set by the Rate Generator, until the specified amount of data has been gathered.

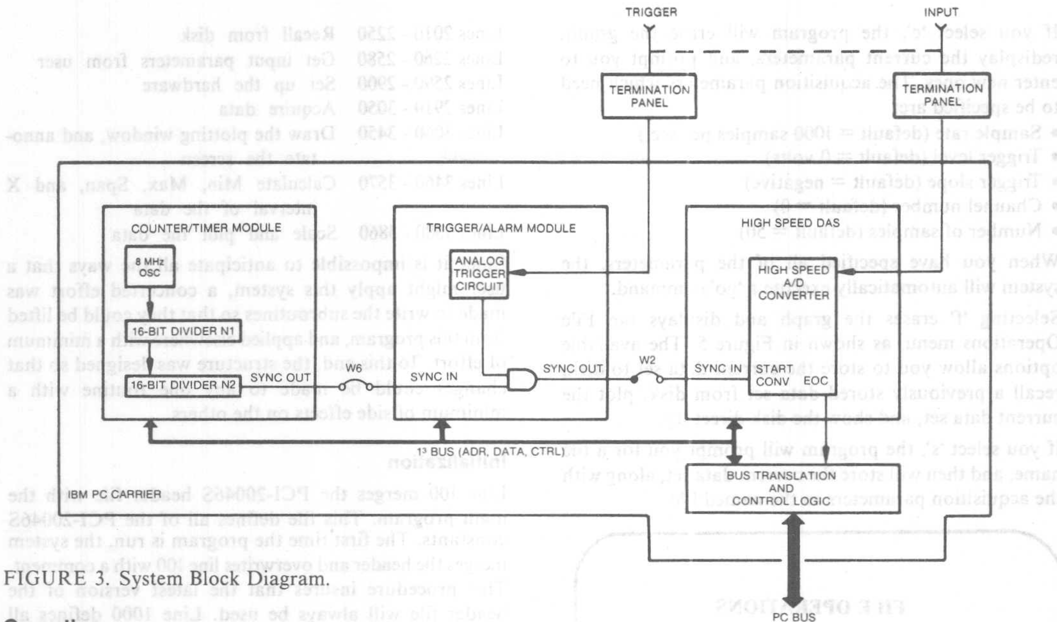


FIGURE 3. System Block Diagram.

Operation

While it is the hardware that defines the performance limits of this type of system, its utility and ease of use are defined by the software (program) which drives it. The program presented here provides all of the basic capabilities that a transient recorder must have: data capture, data display, and store/recall (to/from disk) capabilities for post processing of the data. Many variations on this basic theme are possible. If desired, this program can be used as a model and be easily modified to implement a different approach.

The program makes extensive use of existing machine language subroutines. The subroutines provide a software interface to the hardware, eliminating the requirement that the user 'handle' the multiplexers, sample/holds, A/D converters, etc. These routines are found in the PCI-20046S-1 BASIC Language Modular Software Support Package. The routines are loaded (before executing the main program) by running PCI46S-1.COM, provided on the package's distribution disk. Once installed, the subroutines are accessed via BASIC's CALL statements. Running PCI46S-1.COM will load the support routines into memory, and leave them resident. The base adddress of the routines is stored in the PC at user interrupt vector 60 (hex). This information is used by the program to determine where in memory the routines are located. The "Header" file, which is also provided on the distribution disk, defines the variable names and the other conventions of the subroutine package.

The User Interface

Referring again to Figure 2, we can describe in more detail the user interface portion of the program. Using the single keystrokes indicated on the menu, you can choose to exit the program (quit), start an acquisition using the current parameters (go), change the acquisition parameters (change), or execute a file operation (file).

If you select "q", the program will terminate and return to BASIC.

Selecting 'g' causes the system to first acquire a data set using the defined parameters (if none have been specified, the default parameters will be used). The data is then scaled to fit optimally on the graph, and plotted. The minimum and maximum values of both axes along with the value of the first data sample are labeled (see Figure 4). The program assumes that the A/D module is set up for ±10V full scale. If you choose to jumper the module for another range, remember to modify the related equations accordingly. If desired, the program can be changed to prompt the user for the range being used.

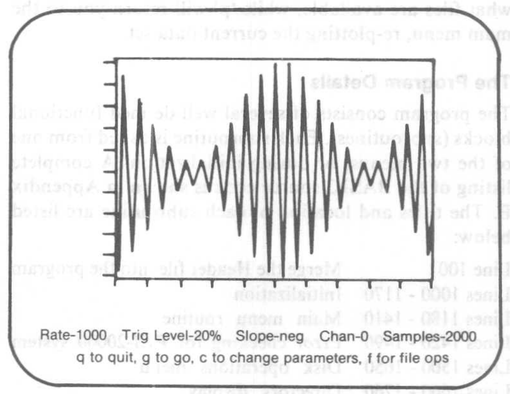


FIGURE 4. Data Display.

If you select 'c', the program will erase the graph, redisplay the current parameters, and prompt you to enter new ones. The acquisition parameters which need to be specified are:

- Sample rate (default = 1000 samples per sec.)
- Trigger level (default = 0 volts)
- Trigger slope (default = negative)
- Channel number (default = 0)
- Number of samples (default = 50)

When you have specified all of the parameters, the system will automatically execute a 'go' command.

Selecting 'f' erases the graph and displays the File Operations menu, as shown in Figure 5. The available options allow you to store the current data set to disk, recall a previously stored data set from disk, plot the current data set, and show the disk directory.

If you select 's', the program will prompt you for a file name, and then will store the current data set, along with the acquisition parameters to the named file.

FILE OPERATIONS

s to save the current data and setup
r to recall stored data and setup
p to plot current data
d for directory listing

****Your choice, please****

FIGURE 5. File Operations Menu.

Selecting 'r' generates a prompt for a file name. The screen is then erased and the new data set, along with its acquisition parameters is plotted.

Selecting 'd' will list the disk directory so you can see what files are available, while 'p' will return you to the main menu, re-plotting the current data set.

The Program Details

The program consists of several well-defined functional blocks (subroutines). Each subroutine is called from one of the two menus, or during initialization. A complete listing of the BASIC source code is shown in Appendix B. The titles and location of each subroutine are listed below:

Line 100	Merge the Header file into the program
Lines 1000 - 1170	Initialization
Lines 1180 - 1410	Main menu routine
Lines 1420 - 1490	Error checking for PCI-20000 system
Lines 1500 - 1650	Disk operations menu
Lines 1660 - 1760	Directory display
Lines 1770 - 2000	Save to disk
Lines 2010 - 2250	Recall from disk
Lines 2260 - 2580	Get input parameters from user
Lines 2590 - 2900	Set up the hardware
Lines 2910 - 3050	Acquire data
Lines 3060 - 3450	Draw the plotting window, and annotate the screen
Lines 3460 - 3570	Calculate Min, Max, Span, and X interval of the data
Lines 3580 - 3860	Scale and plot the data

Since it is impossible to anticipate all the ways that a user might apply this system, a concerted effort was made to write the subroutines so that they could be lifted from this program, and applied elsewhere with a minimum of effort. To this end, the structure was designed so that changes could be made to any one routine with a minimum of side effects on the others.

Initialization

Line 100 merges the PCI-20046S header file with the main program. This file defines all of the PCI-20046S constants. The first time the program is run, the system merges the header and overwrites line 100 with a comment. This procedure insures that the latest version of the header file will always be used. Line 1000 defines all variables not beginning with the letter "f" as integer variables. This is a convenient way to make sure that you don't accidentally attempt to send a floating point number to a routine expecting an integer. All of the calls to the PCI-20046S software routines expect their parameters to be integers.

Line 1040 initializes the PCI-20000 software system, and line 1050 initializes the hardware.

Lines 1120 dimensions the data array used in the program. The maximum number of data points allowed in this version is 5000. This number was selected because interpreted BASIC takes a very long time to process and plot large arrays. If larger arrays are desired it would be a good idea to compile the finished program.

The default values for the data acquisition parameters are set in lines 1130 and 1140. The parameters and values are:

- Sample Rate (FSPS), Initialized to 1000 samples per second
- Trigger level (FTL), Initialized to 0 volts
- Trigger Slope (S$), Initialized to negative (−)
- Channel Number (AI.CHN(1)), Initialized to channel 0
- Number of samples (PASSES), Initialized to 50

Line 1150 calls the routine which sets up the hardware with the defined parameters, and line 1160 calls the routine which draws the window that the plot will appear in. Both of these routines will be discussed in detail later.

The Main Menu Routine

This routine uses the INKEY$ function to accept a single-keystroke input from the user. The available choices are displayed by lines 1220 and 1230. Line 1240

accepts the input and lines 1250 through 1280 determine which operation was requested. If the input was not one of the permitted keys, line 1290 beeps and sends the computer back to line 1240 for another input.

Lines 1340 through 1400 list the subroutines which do the work of setting parameters, acquiring data and displaying data. Two keys can activate these routines. A 'g' will enter at line 1360 and acquire a new data set and plot it using the parameters already in force. A 'c' enters at line 1340 to first get the new parameters. The code then sets up the hardware and executes a data acquisition. Finally the new data is plotted.

The other two menu choices either activate the File Operations Menu (f) or end the program (q).

Getting User-Entered Parameters

If a 'c' is typed from the main menu, the Change subroutine is called. The routine begins at line 2290 by clearing the screen, and by reprinting the current set of parameters for reference. The program then prompts you for each of the parameters required, and checks your inputs against limits. A different user interface could be substituted if desired, so long as when you exit from this routine, the variables FSPS, FTL, S$, AI.CHN(1), and PASSES are all set to the desired values.

Setting Up the System

This is the routine which passes the user-supplied inputs to the hardware for the data-acquisition run. Most of the calls to the PCI-20046S-1 Software Support Package are executed from here.

In lines 2670–2700, the Rate Generator Module is set up to deliver a timing signal at a frequency equal to FSPS. The rate generator consists of an 8MHz oscillator followed by two dividers. COUNT1 and COUNT2 set the divide values and hence set the output frequency to the desired value. FSPS = 8MHz / (COUNT1 * COUNT2). In this program, we have arbitrarily set COUNT2 = 2, so that COUNT1 = 4MHz / FSPS. Since both COUNT1 and COUNT2 can assume only integer values, it is possible that the frequency generated will not be exactly that which was requested. Therefore, in Line 2690, we compute the actual value of FSPS and display it on the main menu.

Line 2700 contains the actual call to the Software Support Package that writes data to the chips on the Rate Generator Module. The format of this call is representative of all the calls used in this program. The keyword CALL tells the computer to jump to a machine language subroutine. The variable name following the call (in this case CNF.RG) defines the starting address of this particular routine within the subroutine library. This offset is specified in the header file which was CHAIN MERGEd at line 100. Any parameters to be passed to the routine are contained in parentheses following the call name.

Lines 2750–2780 set up the Trigger/Alarm Module. This module tests the trigger input signal against both high and low limits. The limits can be programmed to produce a trigger when the input either enters or leaves the 'window' defined by the programmed limits. This establishes the system's trigger level and slope. Because limits are set with eight-bit resolution, a code of zero corresponds to −10V while 255 corresponds to +10 − 1LSB = 9.92V. Line 2750 sets one of the limits to FTL, defining the trigger level. The other limit (S$) is set to 0 for a negative slope or 255 for a positive slope.

Lines 2860–2880 set up the parameters for CNF.HS, the routine which defines the high speed, block mode, analog acquisition. This high-speed routine will scan a channel list specified by the array AI.CHNS. In this program there is only one analog channel. The variables PAC.TYPE, and PAC.CHN specify the signal which is to time the data acquisition. In our case, it will be rate generator channel zero—either the rate generator on the PCI-20007M module, or the built-in pacer clock on the PCI-20041C-2 Carrier, depending on which you are using. The variables TR.TYPE and TR.CHN specify signal which is to trigger the acquisition sequence. In our case, it will be the PCI-20020M-1 Trigger/Alarm Module. The variable HSMODE specifies the type of high speed acquisition to be performed. Setting it to four specifies total hardware control—the fastest possible mode without DMA. In this mode, the Trigger/Alarm Module prevents rate generator pulses from reaching the PCI-20019M until the trigger conditions have been met. When the trigger occurs, A/D conversions are to be initiated directly by hardware. Thus, the system needs to monitor only the A/D's end-of-conversion signal. Each time "EOC" is true, data is stored in memory.

Getting the Data

Line 2960 is the start of the subroutine which acquires the data, while call HS.RUN actually performs the data acquisition. Once triggered, the HS.RUN routine monitors the A/D for an EOC signal and then stores the data in array Y(1), in the order in which it was taken. After the routine has stored the number of values specified by the variable PASSES, it returns control to BASIC. The variable SEGMNT must be set to 0. It will be used for future versions of the software support.

The data in Y(1) is in complementary offset binary format. Line 3030 converts the array into straight offset binary. A value of 0 corresponds to minus full scale (−10V), and a value of 4095 represents plus full scale (10V − 1LSB = 9.9951V).

Drawing the Box

This subroutine establishes a plotting window of defined size, and draws it on the screen. Other routines automatically scale the data to fit within the window.

Lines 3090 and 3100 clear the screen, erase the function key assignments and set the display to the high resolution graphics mode.

Line 3150 defines the size and location of the plotting window. With IBM PCs and compatibles, coordinates 0,0 lie at the upper right hand corner of the screen, while 640,200 is at the lower left. This means that increasing

9

APPLICATIONS, TOOLS & TECHNIQUES
CAPTURING TRANSIENT WAVEFORMS

values of Y move the trace toward the bottom of the screen, which is backwards from what most people would expect. The size and location of the window is defined by specifying the upper right hand corner and the lower left hand corner of the window with variables TOPX,TOPY and BOTX,BOTY. Specifying the corners like this will take care of the inversion of the Y axis. In this program, you can place the window anywhere on the screen simply by changing the values of these variables. However, depending upon the size and location of the window, you may have to move some text to avoid overlap.

Lines 3190—3230 use the LINE command to draw the box. The variable BORD is used to draw the box two pixels larger, in the Y direction, than the actual size of the data window. This assures that if the data is scaled to exactly fit the defined window, it will never be plotted over the box lines themselves.

Lines 3270—3340 draw tic marks on the X and Y axes to form ten evenly spaced intervals for each axis. The size of the tics is adjustable and is set by XTIC and YTIC.

Lines 3380—3450 establish a text line at line 22 to display the acquisition parameters.

Finding Min and Max Data Values

In order to scale the data to fit the plotting window, this routine goes through all the data to find the highest and lowest data values. This is done in lines 3500—3570.

Line 3550 calculates two important parameters: the span of the A/D counts covered by the data, and the number of Y pixels available in the plotting window. The value of FYFACT tells the plotting routine how many Y pixels each A/D count is worth.

Line 3560 calculates how many pixels represent each sample along the Y axis.

Scaling and Plotting The Data

This routine uses the constants computed in the Min/Max routine to autoscale and plot the data. The vertical data to be plotted is in the array Y(I) in the form of A/D counts (integers between 0 and 4095, corresponding to −10V to 9.9951V). To enhance the plot, a line will be drawn from each point in the array to the next. On the time axis, each X coordinate is greater than the previous one by FXINC. The Y increments are computed based upon YSPAN and FYFACT.

Lines 3630—3720 convert the values of the min, max and first data points to volts, and prints them out on the left-hand side of the graph. Line 3960 computes and prints the total elasped time represented by the graph on the right hand side, along the X axis. Lines 3700 and 3710 establish the location of the first data point.

The actual scaling and plotting algorithm occupies lines 3760—3860. For each pass through the loop, the current 'new' X,Y point becomes the 'old' point for the next pass, resulting in a continuous line being drawn through the data points.

The pixel value of the Y coordinate is computed in line 3770. The expression Y(J) − YMIN yields the distance, in A/D counts, from the smallest value in the array to the current data point. This value is multiplied by FYFACT and then added to the offset BOTY to find the correct "Y" pixel location on the physical screen.

The File Operations Menu

The File Operations Menu is displayed when 'f' is selected on the main menu. The routine allows you to type a single character to select one of several tasks: storing data to disk, retrieving data from disk, viewing the disk directory, or replotting the current data set.

Lines 1530—1600 display the menu choices on the screen and prompt you for a response, while line 1610 uses the INKEY$ function to scan the keyboard for an input. When an input is detected, lines 1620—1650 test for a valid choice. Invalid inputs cause the program to go back to the top of the menu.

The Directory Display

Lines 1690—1760 give you the ability to view the currently active directory of either disk drive A or B. The routine prompts you for a single character choice, and then prints the selected directory using the FILES command. If you are using a hard disk with many sub directories, you may want to expand this routine.

Storing Data

This routine stores data to disk. Lines 1830 and 1840 prompt you for a filename, and then create that file. Lines 1960—1980 actually store the data to disk from array Y(I). In lines 1880-1920 the user-defined parameters are also written to the above file. This allows the complete display to be reconstructed when data is recalled from disk by the plotting routine.

The order in which the data is stored is as follows:

PASSES	The number of data points to expect in the file.
FSPS	The sample rate at which the data was taken. We need this in order to label the X axis properly.
S$	The trigger slope. Either 'pos' or 'neg'.
FTL	The trigger level.
AI.CHN(I)	The desired channel number.
Y(I)	The data array. Must contain the number of samples indicated by PASSES.

Recalling Data From Disk

This routine prompts you for an existing filename, and reads the data from disk to RAM. It uses the saved parameters to set up the system as it was when the data was originally taken.

Lines 2110—2150 get the user defined parameters andput them in the appropriate variables. Lines 2190—2210 then write the data points to array Y(I).

Line 2230 calls the setup subroutine which restores the system as directed by the retrieved parameters. At this point, data is in the array Y(I) as if just acquired. Returning to the main menu automatically causes the data to be plotted.

Note that any BASIC program or word processor could have created the data file Y(I), since it is just a list of ASCII characters. This means that data could be entered by hand and then plotted. Alternatively, a program could be written to transfer data from a file created by another data acquisition system. In either case, the data might first have to be manipulated so that the final numbers are in the range from zero to 4095. Also, remember that the parameters must be stored in the order the plotting routine expects, or the system will not function properly.

Converting the Code to Compiled BASIC

Appendix B lists the interpreted BASIC version of this program. While interpreted BASIC is easy to write and debug, it is slow when you need to 'crunch' large amounts of data. For this reason, this version of the program limits the data array to a maximum of 5000 points. At an acquisition rate of 50kHz, it would take 0.1 seconds to acquire the 5000 points of data. However, using interpreted BASIC, on an IBM PC, it would take 5-1/2 minutes to process and plot the data.

The IBM BASIC Compiler can improve the speed of processing by a significant amount. However, the compiler is not 100% compatible with the interpreter. Therefore, some changes must be made to the program before compiling.

Compiled BASIC does not permit the use of the CHAIN MERGE statement. The first change, then, is to merge the file PCIHEAD.BAS with the interpreted version of the program. This header will then be compiled into the program.

The only other changes that need to be made in this example program involve call statements.

The syntax of a call in interpreted BASIC is as follows:

CALL <name>(<param1>,<param2>,...,<paramn>).

For example, the call to the CON.RG routine looks like:

CALL CON.RG (CHN, COUNT1, COUNT2)

where: name is CON.RG, and the parameters are CHN, COUNT1, and COUNT2.

To achieve the same meaning with compiled BASIC we have to use the CALL ABSOLUTE statement. The syntax for this is:

CALL ABSOLUTE (<param1>, <param2>,..., <paramn>, <name>)

Therefore, for each CALL statement in the program we only have to add the keyword, ABSOLUTE, and move the function name to the last item inside the parentheses. Our CON.RG example above, would now appear as:

CALL ABSOLUTE (CHN, COUNT1, COUNT2, CON.RG)

The program is designed such that all the calls to the PCI-20046S-1 Software Support Routines are grouped in three areas: the initialization routine, the system setup subroutine, and the data-taking subroutine. The lines that need to be changed are detailed in Appendix C.

When this program is compiled, the same 5000 data points which took 5-1/2 minutes to process under interpreted BASIC now require only 18 seconds. This is an improvement of better than 18 to one. At this speed, larger arrays make sense, and the compiled version of this program can utilize the full data space of BASIC. Users have successfully dimensioned the data array for 25000 samples.

APPENDIX A

Hardware Configuration

This appendix discusses in detail what data acquisition hardware is required, how to configure the jumpers on the carrier and modules, and how to interconnect the termination panels and cables.

The system is set up to monitor an input in the range of ±10V full scale. For trigger purposes, it will detect when the signal crosses a threshold, and will then initiate digitizing the input signal at a rate of up to 50kHz. The data is automatically stored in RAM. The system uses hardware timing and control to insure the highest possible performance.

The complete list of hardware required is:

PCI-20019M-1 High Speed Data Acquisition Module
PCI-20020M-1 Trigger/Alarm Module

— and either —

PCI-20001C-1 IBM PC Carrier without digital I/O
PCI-20007M-1 Counter/Timer/Rate Generator Module

— or —

PCI-20041C-2 High Performance Carrier.

For convenience in connecting the signals to the modules, termination panels and cables are suggested:

(2) PCI-20057T-1 High Density Analog Termination Panel
(2) PCI-20012A-1 6-ft. Shielded Analog Cable

The Carrier and Modules are configured as follows:

PCI-20001C-1 IBM PC Carrier

Normally, the address switches are set to D0000 (hex). Switches one through six and eight of DIP switch U8 should be in the on position, and switches seven, nine and ten should be off. This establishes the base address of the Carrier card in the PC's memory map. In some circumstances hardware conflicts may require a different address choice. Please refer to the PCI user manual for instructions.

Install the following jumpers:

W6—SYNCOUT of Module position 3 to SYNCIN of Module position 2

W2—SYNCOUT of Module position 2 to SYNCIN of Module position 1

These two jumpers are the mechanism by which clock pulses are passed from the Counter/Timer Module through the Trigger/Alarm Module to the Data Acquisition Module.

Remove all other user selectable jumpers (plug-in type jumpers) on the Carrier.

PCI-20007M-1 Counter/Timer/Rate Generator Module

Install the following jumpers:

W10, W12, W14, W16, W18, W20—All channels software gated.

Remove all other jumpers on the PCI-20007M.

PCI-20041C-2 High Performance Carrier

If you use this carrier, then no PCI-20007M Module is required, since a rate generator is included on the carrier. The base address of the Carrier is normally set at D000 (hex) as is the PCI-20001C. The same cautions about hardware address conflicts apply.

Install the following jumpers:

W27 Pacer out to SYNC BUS
W7 SYNC BUS to SYNCIN Module position 2
W4 SYNCOUT Module position 2 to SYNCIN Module position 3

As with the PCI-20001C, this is the mechanism which routes the clock pulses of the Rate Generator through the Trigger/Alarm Module to the Data Acquisition Module. All other plug-in jumpers should be removed on the PCI-20041C-2.

PCI-20020M-1 Trigger/Alarm Module

Install the following jumpers:

W7, W9 Select single input from the termination panel.
W19 SYNCIN gated to SYNCOUT when trigger is true.

Remove all other jumpers on the PCI-20020M.

PCI-20019M-1 High Speed Data Acquisition Module

Install the following jumpers:

W2, W4 Set the input range to ±10V.
W8, W9 Start conversion on rising edge of SYNCIN.
W11 Single channel mode (disable channel scan).

Remove all other jumpers on the PCI-20019M.

After all the jumpers are set as outlined above, the Modules are installed in the Carrier as follows if the PCI-20001C-1 is used:

- Install the PCI-20019M High Speed DAS in J1 on the Carrier
- Install the PCI-20020M Trigger/Alarm in J2 on the Carrier
- Install the PCI-20007M Rate Generator in J3 on the Carrier.

Plug the other end of the cable coming from the DAS Module into P1 of the PCI-20057T-1 High Density Analog Termination Panel. Likewise, plug the other end of the cable connected to the Trigger/Alarm Module into P2 of the termination panel. Connect your input signal to any of the terminals marked zero through seven, and the ground return to any of the ground terminals in the 'Group 1' section of the termination panel.

Normally, the input signal also serves as the trigger. In this case a connection should be made on the termination panel from the selected analog input to the input of the Trigger Module. If an external event is to be used as a trigger, then that signal can be connected to the trigger input. Any signal in the range of ±10V will do. If you are triggering from the input signal, run a jumper from your input channel to terminal eight in the 'Group 2' section of the termination panel. This is the input to the Trigger/Alarm Module. If you are triggering from an external source, connect the external signal here. The hardware is now set up and ready to acquire data.

If the PCI-20041C-2 is used:

- Install the PCI-20019M High Speed DAS in J3 on the Carrier
- Install the PCI-20020M Trigger/Alarm in J2 on the Carrier.

J1 is the connector nearest to the IBM PC bus connector. After the Modules are plugged in, insert the assembly into the computer (or expansion chassis if you are using one.)

It is suggested that the accessory Strain Relief Bracket (PCI-20028A-2) to be used to facilitate making cable connections. In this case, run the cables out through the 'open' rear port of the PC, next to the Carrier. Plug the cables onto the Trigger/Alarm and DAS Modules. Make sure that the end marked Computer goes to the Modules, otherwise the shielding will not be connected properly. Orient the connectors such that the cable naturally goes from the module toward the rear of the computer, without 'doubling back.'

APPENDIX B

Transient Capture System Program Listing

```
100 CHAIN MERGE "PCIHEAD.BAS", 100, ALL
1000 DEFINT A-E,G-Z
1010 '
1020  ' Initialize the hardware and software system
1030  '
1040      CALL SYSINIT
1050      SEGMT = &HD000
```

```
1060     CALL INIT(SEGMT)
1070 GOSUB 1470  ' check for init errors
1080  '
1090  ' This section of the program sets all the default
1100  ' parameters, and configures the system
1110 '
1120 DIM Y(5000) : EN = 1 : DIM AI.CHN(2)
1130 FSPS = 1000 : FTL = 0 : S$ = "-"
1140 AI.CHN(1) = 0 : PASSES = 50
1150 GOSUB 2590 'set up system
1160 GOSUB 3090 'draw box
1170  '
1180  ' Main menu routine. All action is initiated from here. The
1190  ' various setup, acquire, and plot routines are all arranged
1200  ' as a group of subroutines called from here.
1210  '
1220  LOCATE 24,12 : PRINT " q to quit, g to go, c to change";
1230  PRINT " parameters, f for file ops ";
1240  P$ = INKEY$ : IF P$ = "" THEN 1240
1250  IF P$ = "g" OR P$ = "G" THEN 1360
1260  IF P$ = "c" OR P$ = "C" THEN 1340
1270  IF P$ = "q" OR P$ = "Q" THEN 1410
1280  IF P$ = "f" OR P$ = "F" THEN 1530
1290  BEEP : GOTO 1240    ' an illegal character was typed.
1300 '
1310 ' These are the subroutine calls which actually do the
1320 '  work.
1330 '
1340  GOSUB 2290 ' get parameters
1350  GOSUB 2590 ' set up system
1360  GOSUB 2910 ' get the data
1370  GOSUB 3090 ' draw the box
1380  GOSUB 3500 ' find min/max
1390  GOSUB 3630 ' plot the data
1400  GOTO 1180
1410 SCREEN 0,0,0 : END
1420  ' Subroutine to check for errors in the pci20k calls.
1430  ' This call is executed after every pci20k call.
1440  ' All we do is check for nonzero error codes, print them out,
1450  ' and abort. One could be more imaginative if desired.
1460  '
1470 CALL ERR.SYS(Z) :
1480 IF Z <>0 THEN PRINT "Error  ";Z : GOTO 1410
1490 RETURN
1500  '
1510  ' File operation submenu
1520  '
1530 CLS : LOCATE 8, 30 : PRINT "FILE OPERATIONS"
1540 PRINT
1550 P$ = ""
1560 PRINT "s to save the current data and setup"
1570 PRINT "r to recall stored data and setup"
1580 PRINT "p to plot current data"
1590 PRINT "d for directory listing"
1600 LOCATE 15,25 :PRINT " ** Your choice, please **  "
1610 P$ = INKEY$ : IF P$ = "" THEN 1610
1620 IF P$ = "s" OR P$ = "S" THEN 1770
1630 IF P$ = "r" OR P$ = "R" THEN 2010
1640 IF P$ = "p" OR P$ = "P" THEN GOSUB 2590 : GOTO 1370
1650 IF P$ <> "d" AND P$ <> "D" THEN 1530
1660 '
1670 ' If we got here, the file operation choice was "d" for directory
1680 '
1690 INPUT "Disk  ( a or b ) -- ",D$
1700 IF D$ = "a" THEN GOTO 1720
1710 GOTO 1730
1720 FILES"a:*.*"
1730 IF D$ = "b" THEN FILES"b:*.*"
1740 PRINT: PRINT " Hit any key to continue"
1750 IF INKEY$ = "" THEN 1750
1760 GOTO 1530
1770  '
1780  ' Save data to disk routine. The data is written to disk in
1790 ' ASCII form. The first thing stored is the number of data points
1800 ' in the file. The next four items are the rest of the user
1810 ' entered parameters. Finally all the data points are stored.
1820  '
1830 CLS : INPUT "Filename to write data to -- ",FILNAM$
1840 OPEN FILNAM$ FOR OUTPUT AS #1
1850 '
1860 ' Store the user entered parameters
1870 '
```

```
1880 PRINT #1,PASSES
1890 PRINT #1, FSPS
1900 PRINT #1, S$
1910 PRINT #1, FTL
1920 PRINT #1, AI.CHN(1)
1930 '
1940 ' Store the data
1950 '
1960 FOR I = 1 TO PASSES
1970 PRINT #1, Y(I)
1980 NEXT I
1990 CLOSE #1
2000 GOTO 1530
2010 '
2020 ' Recall data from disk routine. The data must be retrieved
2030 ' in the same order in which it was stored.
2040 '
2050 '
2060 CLS : INPUT "Filename to recall -- ",FILNAM$
2070 OPEN FILNAM$ FOR INPUT AS #1
2080 '
2090 ' Retrieve the user entered parameters
2100 '
2110 INPUT #1, PASSES
2120 INPUT #1, FSPS
2130 INPUT #1, S$
2140 INPUT #1, FTL
2150 INPUT #1, AI.CHN(1)
2160 '
2170 ' Retrieve the data
2180 '
2190 FOR I = 1 TO PASSES
2200 INPUT #1, Y(I)
2210 NEXT I
2220 CLOSE #1
2230 GOSUB 2590
2240 CLOSE #1
2250 GOTO 1530         ' Return to file menu
2260 '
2270 ' Subroutine to get acquisition parameters from the user
2280 '
2290 CLS : GOSUB 3380  '  Clear screen and print current params
2300 LOCATE 1,1
2310 '
2320 ' Get sample rate
2330 '
2340 INPUT "Samples per second (150 - 50000) ------- ",FSPS
2350 IF FSPS < 150 OR FSPS > 50000! THEN 2290
2360 '
2370  ' Get trigger level
2380  '
2390 INPUT "Trigger level (+/- %FS) --------------- ", FTL
2400 IF FTL > 100 OR FTL <-100 THEN 2390
2410 '
2420 ' Get trigger slope
2430 '
2440 INPUT "Trigger slope ( + or -) --------------- ",S$
2450 IF S$ = "+" THEN  S$ = "pos"
2460 IF S$ = "-" THEN  S$ = "neg"
2470 IF S$ <> "pos" AND S$<> "neg" THEN 2440
2480  '
2490  ' Select the input channel. Must be in the range of 0 - 7
2500  '
2510 INPUT "Input channel (0 - 7) ----------------- ", AI.CHN(1)
2520 IF AI.CHN(1) > 7 OR AI.CHN(1) < 0 THEN 2510
2530 '
2540 ' Get number of passes
2550 '
2560 INPUT "Number of samples (10 - 5000) ------- ", PASSES
2570 IF PASSES >5000 OR PASSES < 10 THEN GOTO 2560
2580 RETURN
2590 '
2600  ' Subroutine to set up system to selected parameters
2610 '
2620 ' Configure rate generator for correct scan rate
2630 ' Since not all scan rates can be hit exactly, we will get as
2640 ' close as possible, and then report the actual rate we got.
2650 ' The rate generator mode will be 2
2660 '
2670 FMEG4 = 4000000! : FMEG8 = 8000000! : COUNT1 = FMEG4 / FSPS
2680 COUNT2 = 2 : CHN = 0 : FCNT1 = COUNT1 : FCNT2 = COUNT2 : RGMODE = 2
2690 FSPS = FMEG8 / (FCNT1 * FCNT2) ' recompute the actual sample rate
```

```
2700 CALL CNF.RG(CHN, COUNT1, COUNT2, RGMODE)
2710 GOSUB 1470           ' Test for errors after every call
2720 '
2730 ' Set up the Trigger/Alarm module for the level and slope
2740 '
2750 LEVEL1 = ((FTL / 100)) * 128 + 127 : WINDW = 0
2760 IF S$ = "pos" THEN  LEVEL2 = &HFF
2770 IF S$ = "neg" THEN  LEVEL2 = 0
2780 CALL CNF.TRIG(CHN, LEVEL1, LEVEL2, WINDW)
2790 GOSUB 1470
2800 '
2810 ' Set up the channel list for the high speed acquisition,
2820 ' and execute the configure instruction. We will use mode 4 -
2830 ' hardware controlled acquisition. Trigger will be the PCI-20020M,
2840 ' and acquisition will be paced by the PCI-20007M
2850 '
2860 AI.CHN(2) = -1 : PACER = RG : PCHN = 0 : HSMODE = 4
2870 TTYP = TRIG : TCHN = 0
2880 CALL CNF.HS(PACER, PCHN, HSMODE, TTYP, TCHN, AI.CHN(1))
2890 GOSUB 1470
2900 RETURN
2910  '
2920  ' Execute the high speed acquisition.
2930  ' This is the routine which actually takes the data. All
2940  ' parameters have been set elsewhere.
2950  '
2960 SEGMNT = 0
2970 PRINT "running";
2980 CALL HS.RUN(PASSES, Y(1), SEGMNT)
2990 '
3000 ' The DAS module's code is complimentary binary, so invert the data
3010 ' to get sensible numbers
3020 '
3030 FOR I = 1 TO PASSES : Y(I) = Y(I) XOR &HFFF : NEXT I
3040 GOSUB 1470
3050 RETURN
3060 '
3070 ' Subroutine to draw the plotting window
3080 '
3090 CLS : KEY OFF ' Clear the screen and keys
3100 SCREEN 2  ' Select high resolution graphics
3110 '
3120 ' Define the location and size of the graph.  The upper left hand
3130 ' is at topx,topy and the lower right at botx,boty.
3140 '
3150 TOPX = 600 : BOTX =56 : TOPY = 5 : BOTY = 145
3160
3170 ' Draw the box with a clear border of 2 pixels at top and bottom.
3180 '
3190 BORD = 2
3200 LINE (BOTX,BOTY+BORD) - (BOTX,TOPY-BORD)
3210 LINE (BOTX,BOTY+BORD) - (TOPX, BOTY+BORD)
3220 LINE (TOPX,TOPY-BORD) - (TOPX,BOTY+BORD)
3230 LINE (TOPX,TOPY-BORD) - (BOTX,TOPY-BORD)
3240 '
3250 ' Divide the x and y axes into 10 intervals, and draw tic marks
3260 '
3270 FXINT = (TOPX - BOTX) / 10 : FYINT = (TOPY - BOTY) / 10
3280 YTIC = 5 : XTIC = 10
3290 FOR I = 0 TO 10
3300 LINE (BOTX, BOTY + I * FYINT) - (BOTX - XTIC, BOTY + I * FYINT)
3310 NEXT I
3320 FOR I = 0 TO 10
3330 LINE (BOTX + I * FXINT, BOTY) - (BOTX + I * FXINT, BOTY + YTIC)
3340 NEXT I
3350 '
3360 ' Print out the acquisition parameters below the graph
3370 '
3380 TEXTLIN = 22
3390 LOCATE TEXTLIN, 5
3400 PRINT "Rate --";FSPS;" ";
3410 IF S$ = "-" THEN S$ = "neg"
3420 IF S$ = "+" THEN S$ = "pos"
3430 PRINT "Trig Level --";FTL;"%  Slope -- ";S$;"  Chan --";AI.CHN(1);
3440 PRINT " Samples --";PASSES
3450 RETURN
3460 '
3470 'Subroutine to find the min and max data values
3480 '  and calculate the span and x interval of the data points
3490 '
3500 YMIN = 4095 : YMAX = 0
```

```
3510 FOR I = 1 TO PASSES
3520 IF Y(I) > YMAX THEN YMAX = Y(I)
3530 IF Y(I) < YMIN THEN YMIN = Y(I)
3540 NEXT I
3550 YSPAN = YMAX - YMIN : FYFACT = (TOPY - BOTY) / YSPAN
3560 FXINC = (TOPX - BOTX) / (PASSES - 1)
3570 RETURN
3580 '
3590 ' Subroutine to label the axes and scale and plot the data
3600 '
3610 ' First, label the x and y axes, and the first data value
3620 '
3630 FLO = (YMIN/4096) * 20 -10 ' Label the y axis
3640 FHI = (YMAX/4096) * 20 -10
3650 FTRIG = (Y(1)/4096) * 20 - 10
3660 LOCATE 1,BOTX/8-6 : PRINT USING "###.##"; FHI
3670 LOCATE BOTY/8+1,BOTX/8-6 : PRINT USING "###.##"; FLO
3680 LOCATE BOTY/8 + 2, BOTX/8: PRINT"0"; ' Label the X axis
3690 LOCATE BOTY/8 + 2, TOPX/8-3 : PRINT USING "##.####"; PASSES/FSPS
3700 FOLDX = BOTX-FXINC
3710 OLDY = (Y(1) - YMIN) * FYFACT + BOTY
3720 LOCATE OLDY/8+1 ,BOTX/8-6 : PRINT USING "###.##"; FTRIG
3730  '
3740  'Scale and plot the data
3750  '
3760 FOR J = 1 TO PASSES
3770 NEWY = (Y(J) - YMIN) * FYFACT + BOTY
3780 FXNEW = FOLDX + FXINC
3790 '
3800 ' Special adjustment to put the first point on the graph
3810 '
3820 IF FOLDX < BOTX THEN FOLDX = BOTX
3830 LINE (FOLDX, OLDY) - (FXNEW, NEWY)
3840 FOLDX = FXNEW : OLDY = NEWY
3850 NEXT J
3860 RETURN
```

APPENDIX C

This appendix details the differences between the Compiled and Interpreted versions of the program. The Compiler used was the IBM Personal Computer BASIC Compiler, Version 2.00.

First, PCIHEAD.BAS must be merged into the program, and then the following lines changed as shown: (Note that call to AUTOGRPH on line 840 is in the header file, so the line number can vary depending on how you have initialized your software support package.)

```
840  CALL ABSOLUTE (VERS.L, VCHK(1), VCHK.L, AUTOGRPH)
1040 CALL ABSOLUTE (SYSINIT)
1060 CALL ABSOLUTE (SEGMT, INIT)
1120 DIM Y (25000) : EN = 1 : DIM AI.CHN(2)
1470 CALL ABSOLUTE (Z, ERR.SYS)
2560 INPUT "Number of samples (10 - 25000) --------- ", PASSES
2570 IF PASSES > 25000 OR PASSES < 10 THEN GOTO 2560
2700 CALL ABSOLUTE (CHN, COUNT1, COUNT2, RGMODE, CNF.RG)
2780 CALL ABSOLUTE (CHN, LEVEL1, LEVEL2, WINDW, CNF.TRIG)
2880 CALL ABSOLUTE (PACER, PCHN, HSMODE, TTYP, TCHN, AI.CHN(1), CNF.HS)
2980 CALL ABSOLUTE (PASSES, Y(1), SEGMNT, HS.RUN)
```

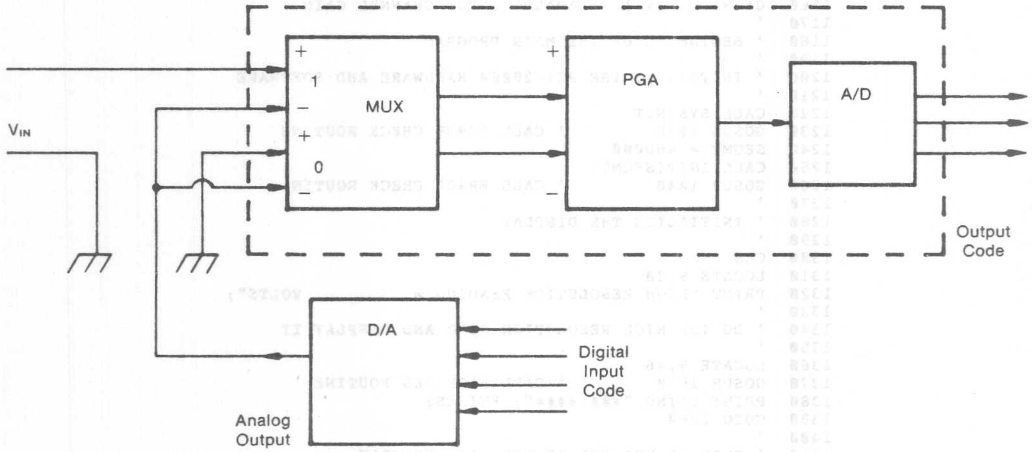
As can be seen from the above, three essential changes have been made. First, the header file PCIHEAD has been merged into the program prior to compilation. Second, all CALL statements have been changed to CALL ABSOLUTE. Finally, the array Y(I), size has been changed from 5000 elements to 25000 elements since the greater processing speed of compiled BASIC allows us to effectively use more data.

GETTING INCREASED RESOLUTION FROM A 12-BIT DATA ACQUISITION MODULE

Using IBM PC, PC/XT, PC/AT and Compatible Computers

INTRODUCTION

It is possible to "make" a high resolution (16 bits) A/D converter from a 12-bit unit. This technique uses readily available PCI-20000 series modules including a 12-bit A/D (PCI-20002M-1) module and a standard 16-bit D/A (PCI-20006M-1) module. Conceptually, increased resolution is achieved by performing two conversions. The first conversion is done in a low gain (1 or 10). The DAC is then set to "buck-out" or "offset" the original reading. At this point the gain is changed to 1000, magnifying the difference between the first reading and the actual input voltage. With a gain increase of 100 (from 10 to 1000), more than 6 bits of resolution are added ($2^6 = 64$). The second conversion records this additional data. Simple math sums the two readings together, while accounting for the gain changes and channel offsets. In general terms the resulting resolution and accuracy are determined by the 16 bit D/A. This usually yields 16 bit resolution with greater than 14 bit accuracy. The example programs below implement hardware error checking, offset error correcting and the resolution enhancing algorithm. The resulting A/D has up to seven (7) Single-Ended inputs, and can measure DC or slowly changing signals.



Functional Block Diagram Of High Resolution A/D Converter

Notes: The three top blocks (Mux, PGA & A/D) represent parts of the PCI-20002M-1 A/D module.

While the A/D module is in the Differential mode, the signal input is Single-Ended.

The D/A is a 16 bit analog output (PCI-20006M-1) module.

The D/A output can be connected to the PCI-20002M-1 inputs through the internal PCI-20000 Data Acquisition Bus (I^3 Bus). External jumpers are not required.

The measurement algorithm is as follows:

1. Set the D/A output to zero and measure the input signal on channel 0 with the PGA in a gain of 1000.
2. Store this result (offset voltage).
3. Switch to channel 1 (or other active channel) and measure Vin in a gain of 1 or 10 (Gain = 100 can be used, but the resulting resolution will be about 15 bits vs 16 bits). Store this result and set the D/A to this value.
4. Raise the PGA gain to 1000.
5. Measure the "new" input signal ($V_{In} - V_{First\ Conversion}$). Store this result.
6. Add the 1st reading to the 2nd reading, taking into account the gain change and the offset voltage. The final measurement value is determined by averaging several readings to reduce the influence of noise.

9

APPLICATIONS, TOOLS & TECHNIQUES
HIGH RESOLUTION A/D

LISTING 1 - BASIC LANGUAGE PROGRAM

```
10    '
20    ' BASIC PROGRAM FOR OBTAINING 16 BIT RESOLUTION
30    '   FROM A 12 BIT ANALOG TO DIGITAL CONVERTER
40    '
50    ' The System Requires:   PCI-20002M-1 A/D Module at +/- 10 V Diff.
60    ' -------------------    PCI-20006M-1 D/A Module at +/- 10 V
70    '                        PCI-20046S-1 Software Drivers
80    '
90    CHAIN MERGE "PCIHEAD.BAS", 100, ALL
1000  GOTO 1090
1010  '
1020  ' DEFINE AN ERROR CHECKING ROUTINE
1030  '
1040  CALL ERR.SYS(ERROR.NO)
1050  IF ERRORNO = 0 THEN GOTO 1080
1060  PRINT "ERROR:  ";ERRORNO
1070  END
1080  RETURN
1090  '
1100  ' DEFINE SOME APPLICATION CONSTANTS
1110  '
1120  AICHAN    = 1  ' ANALOG INPUT CHANNEL
1130  AIREFCHAN = 0  ' ANALOG REFERENCE CHANNEL
1140  DACCHAN   = 0  ' 16 BIT DAC CHANNEL
1150  AVG1000   = 10 ' NUM TIMES TO AVERAGE READINGS AT GAIN=1000
1160  GAIN1     = 1  ' ANALOG INPUT CHANNEL GAIN
1170  '
1180  ' BEGINNING OF THE MAIN PROGRAM
1190  '
1200  ' INITIALIZE THE PCI-20000 HARDWARE AND SOFTWARE
1210  '
1220  CALL SYSINIT
1230  GOSUB 1040           ' CALL ERROR CHECK ROUTINE
1240  SEGMT = &HD000
1250  CALL INIT(SEGMT)
1260  GOSUB 1040           ' CALL ERROR CHECK ROUTINE
1270  '
1280  ' INITIALIZE THE DISPLAY
1290  '
1300  CLS
1310  LOCATE 9,20
1320  PRINT "HIGH RESOLUTION READING =          VOLTS";
1330  '
1340  ' DO THE HIGH RESOLUTION READ AND DISPLAY IT
1350  '
1360  LOCATE 9,46
1370  GOSUB 1610           ' CALL HIGH RES ROUTINE
1380  PRINT USING "+##.####"; FHIRES;
1390  GOTO 1360
1400  '
1410  ' THIS IS THE END OF THE MAIN PROGRAM
1420  '
1430  ' HIGH RESOLUTION READING ROUTINE
1440  '
1450  ' This routine reads the voltage on AICHAN using the analog
1460  ' output DACCHAN, and the analog input AIREFCHAN.  The system
1470  ' must be jumpered as follows:
1480  '
1490  '    * PCI-20002M-1    +/- 10 V, Differential
1500  '    * PCI-20006M-1    +/- 10 V
1510  '
1520  ' The method sets DACCHAN to 0 volts, reads Vin with AICHAN
1530  ' at a gain of GAIN1, sets DACCHAN to near that voltage, reads
1540  ' the difference between DACCHAN's voltage and Vin at a gain
1550  ' of 1000, and adds the two together to get the AICHAN voltage
1560  ' measurement.  AIREFCHAN is used to measure and correct for
1570  ' offset in the ADC and DAC circuits.
1580  '
1590  ' SET THE DAC OUTPUT TO 0 VOLTS
1600  '
1610  AOCHN   = DACCHAN
1620  AOCOUNT = 0
1630  CALL WRITE.CH(AO, AOCHN, AOCOUNT)
1640  GOSUB 1040           ' CALL ERROR CHECK ROUTINE
1650  '
1660  ' READ REFERENCE CHANNEL FOR SYSTEM OFFSET
1670  '
```

```
1680 AIREFCHN = AIREFCHAN
1690 GAIN     = 1000
1700 ZCHN     = -1
1710 RANGE    = 1
1720 CALL CNF.AI(AIREFCHN, GAIN, ZCHN, RANGE)
1730 GOSUB 1040          ' CALL ERROR CHECK ROUTINE
1740 '
1750 FOFFSETSUM = 0!
1760 FOR I = 1 TO AVG1000
1770   CALL READ.CH(AI, AIREFCHN, ADCVAL)
1780   GOSUB 1040        ' CALL ERROR CHECK ROUTINE
1790   FOFFSETSUM = FOFFSETSUM + ADCVAL
1800 NEXT I
1810 '
1820 ' READ AICHAN FOR INITIAL 12 BIT READING
1830 '
1840 AICHN    = AICHAN
1850 GAIN     = GAIN1
1860 CALL CNF.AI(AICHN, GAIN, ZCHN, RANGE)
1870 GOSUB 1040          ' CALL ERROR CHECK ROUTINE
1880 CALL READ.CH(AI, AICHN, AICOUNT1)
1890 GOSUB 1040          ' CALL ERROR CHECK ROUTINE
1900 '
1910 ' CALCULATE AND SET DACCHAN TO NEAR VIN
1920 '
1930 AOCOUNT = 16 * (AICOUNT1 - 2048) / GAIN1
1940 CALL WRITE.CH(AO, AOCHN, AOCOUNT)
1950 GOSUB 1040          ' CALL ERROR CHECK ROUTINE
1960 '
1970 ' READ DIFFERENTIAL VOLTAGE AT A GAIN OF 1000
1980 '
1990 GAIN = 1000
2000 CALL CNF.AI(AICHN, GAIN, ZCHN, RANGE)
2010 GOSUB 1040          ' CALL ERROR CHECK ROUTINE
2020 '
2030 FAICOUNTSUM = 0!
2040 FOR I = 1 TO AVG1000
2050   CALL READ.CH(AI, AICHN, ADCVAL)
2060   GOSUB 1040        ' CALL ERROR CHECK ROUTINE
2070   FAICOUNTSUM = FAICOUNTSUM + ADCVAL
2080 NEXT I
2090 '
2100 ' CALCULATE MEASURED VOLTAGE
2110 '
2120 FHIRES1 = ((AICOUNT1/4096!*20!)-10!)/GAIN1
2130 FHIRES2 = (((FAICOUNTSUM/AVG1000)/4096!*20!)-10!)/1000!
2140 FHIRES3 = (((FOFFSETSUM/AVG1000)/4096!*20!)-10!)/1000!
2150 FHIRES  = FHIRES1 + FHIRES2 - FHIRES3
2160 '
2170 RETURN
```

LISTING 2 - TURBO-PASCAL LANGUAGE, FUNCTION DEFINITION

```
{ NOTE :  This function requires the PCI-20046S-3 software drivers.
          The system must be jumpered as follows :

             *  PCI-20002M-1              +/- 10 volts, differential
             *  PCI-20006M-1 or -2        +/- 10 volts
  The function sets DACChan to 0 volts, reads Vin with ADCChan at a
  gain of one, sets DACChan to near that voltage, reads the difference
  between DACChan's voltage and Vin at a gain of 1000, and adds the
  two togethere to get the ADCChan voltage measurement.  ADCRefChan is
  used to measure and correct for offset errors in the ADC and DAC
  circuits.
}
Function HiResADC(ADCChan, ADCRefChan, DACChan : Integer) : Real ;
Const
  Gain1    = 1  ;            { Gain for first reading          }
  Avg1000  = 10 ;            { Averaging passes; Gain = 1000   }
  Avg1     = 1  ;            { Averaging passes; gain = 1      }
Var
  AICount1,                  { First ADC reading at Gain1      }
  Vmeas,                     { Measured Voltage                }
  SysOffsetSum,              { System offset counts            }
  AICountSum    : Real ;     { ADC readings at gain of 1000    }
  { CheckError -- check for PCI-20046S-3 system error. }
  Procedure CheckError ;
```

```
  Var
    Error : Integer ;
  Begin  { CheckError }
    Error := ErrSys ;
    If Error <> 0 Then
    Begin
      WriteLn('ERROR :  ', Error) ;
      Halt
    End ;
  End ;  { CheckError }
  { SetDAC - sets the DAC output voltage. }
  Procedure SetDAC(DACCount : Integer) ;
  Begin  { SetDAC }
    WriteCh(_AO, DACChan, DACCount) ;
    CheckError                      ;
  End ;  { SetDAC }
  { ReadADC - reads the ADC. }
  Function ReadADC(ADCChannel, Gain, Passes : Integer) : Real ;
  Var
    ZChn,
    Range,
    Index     : Integer ;
    CountSum : Real     ;
  Begin  { ReadADC }
    ZChn  := -1 ;
    Range :=  1 ;
    CnfAI(ADCChannel, Gain, ZChn, Range) ;
    CheckError                           ;
    CountSum := 0.0 ;
    For Index := 1 to Passes Do
    Begin
      CountSum := CountSum + ReadCh(_AI, ADCChannel) ;
      CheckError                                     ;
    End ;
    ReadADC := CountSum / Passes ;
  End ;  { ReadADC }
  Begin  { HiResADC }
    { ***** Set the DAC for a voltage of 0 ***** }
    SetDAC(0)
    { ***** Read ADCRefChan to get the system offset voltage ***** }
    SysOffsetSum := ReadADC(ADCRefChan, 1000, Avg1000) ;
    { ***** Read ADCChan at a gain of Gain1 for Vin ***** }
    AICount1 := ReadADC (ADCChan, Gain1, Avg1) ;
    { ***** Write DACChan with near Vin voltage ***** }
    SetDAC(Round((AICount1-2048.0)*16.0/Gain1)) ;
    { ***** Read ADCChan at a gain of 1000 for near 0 voltage ***** }
    AICountSum := ReadADC(ADCChan, 1000, Avg1000) ;
    { ***** Calculate measured Vin ***** }
      { ***** Calculate low resolution voltage component ***** }
      Vmeas := ((AICount1/4096.0*20.0)-10.0)/Gain1 ;
      { ***** Calculate and add high resolution voltage component ***** }
      Vmeas := Vmeas + (((AICountSum/4096.0*20.0)-10.0)/1000.0) ;
      { ***** Calculate and subtract system offset error component ***** }
      Vmeas := Vmeas - (((SysOffsetSum/4096.0*20.0)-10.0)/1000.0) ;
    { ***** Return measured Vin ***** }
    HiResADC := Vmeas ;
  End ; { HiResADC }
```

LISTING 3 – TURBO-PASCAL LANGUAGE PROGRAM

```
  { EXAMPLE USE OF HiRes.pas Function }
  Program HiResDmo
  {$I PCIHEAD.PAS  }          { Include PCI-20046S-3 Support   }
  {$I P46.PAS      }          { Include PCI-20046S-3 Interface }
  {$I HiResadc.PAS }              { Include HiRes Function }
    { CheckError -- check for PCI-20046S-3 system error. }
    Procedure CheckError ;
    Var
      Error : Integer ;
    Begin { CheckError }
      Error := ErrSys ;
      If Error <> 0 Then
      Begin
        WriteLn('ERROR :  ',Error) ;
        Halt                       ;
      End ;
    End ; { CheckError }
```

```
Const
   ADCChan    = 1 ;
   DACChan    = 0 ;
   ADCRefChan = 0 ;
Var
   Vect
   Segment : Integer ;
Begin  { HiResDmo }
   Vect := $60  ;
   SetVec(Vect) ;
   Sysinit      ;
   CheckError   ;
   ClrScr ;
   GotoXY(20,9) ;
   Write('HIGH RESOLUTION READING =           Volts') ;
   Segment := $D000 ;
   Init (Segment)   ;
   CheckError       ;
   Repeat
      GotoXY(46,9)       ;
      Write('          ') ;
      GotoXY(46,9)       ;
      Write(HiResADC(ADCChan, ADCRefChan, DACChan):8:4) ;
   Until KeyPressed ;
End.    { HiResDmo }
```

PROGRAMMING YOUR PC FOR DATA ACQUISITION APPLICATIONS USING IBM PC, PC/XT, PC/AT AND COMPATIBLE MACHINES

SOME USEFUL TECHNIQUES

- Establishing a Time-Base from the PC's clock
- Time-Stamping
- Converting the raw data to a LOTUS readable file
- Writing the data to disk

INTRODUCTION

In most data acquisition tasks there are three basic steps :

1) Acquire the data
2) Process, manipulate or analyze the data
3) Present the data in tabular, graphical or report format.

Using the PCI-20000 System, extensive programming experience is not needed to acquire or manipulate data. This Application Note will demonstrate how to read a given Channel at predetermined intervals and to write a permanent record of this data to disk. As is often required, the data will also be "Time-Stamped." This means that the data will be recorded along with the corresponding Time-of-Day.

Manipulating and presenting the data can get "involved" and as a result, may require considerable effort. Fortunately, a number of software products exist that bridge the gap between the raw data and sophisticated analysis and presentation capabilities. One such package is LOTUS SYMPHONY™. SYMPHONY integrates word-processing and spreadsheet functions. The spreadsheet portion is called LOTUS 1-2-3™, and it provides both analysis and graphics features. This note shows how easy it is to transform data files into a format that can be read by LOTUS. The technique shown is general and can be applied to any language. An example is included using BASIC.

EXAMPLE

In the example that follows, a PCI-20000 system consisting of a PCI-20001C-1 Carrier and a PCI-20002M-1 Data Acquisition Module is used to acquire analog input data and place it into an array. The complete program is shown in listing 1. The first portion of the program makes use of the PCI-20046S-1 Software drivers. These drivers "isolate" the programmer from having to have a complete understanding of the data acquisition hardware. They handle the details of setting multiplexers, programmable gain amplifiers, sample-holds and A/D converters. Programming READ, WRITE and many other functions are reduced to invoking simple "CALL" statements. In many cases, it is required that data be taken at defined, regular intervals. The sample program includes both a time-base generator and a time-stamping routine. The maximum data acquisition rate, using this technique, is limited by the timing jitter generated by the PC. This timing uncertainty is about 10 milliseconds, therefore, the practical maximum data rate might be about 1 reading per second (1% timing accuracy). Having acquired the data, the program writes the data to a LOTUS readable file. Listing 2 is an example of such a file. This data can be presented graphically by LOTUS as shown in figure 1.

BACKGROUND

In this example, two BASIC functions are used to implement the timing and time-stamping features. They are TIMER and TIME$, respectively. TIMER returns the number of seconds elapsed since system start-up, or since last midnight if a real-time clock is installed in the computer. TIME$ returns TIMER's value formatted in "hh:mm:ss".

Getting data into the LOTUS format is actually very straight forward. There are only a few simple rules to remember:

RULE 1 :	The data file must have a .PRN extension.
RULE 2 :	String type (literal) information must be contained in quotes (" "). Numbers (non-literals) must not be contained in quotes.
RULE 3 :	Entries that are to be on the same row in LOTUS 1-2-3, must be separated by commas in the file.
RULE 4 :	Each line must end with Carriage-Return, Line-Feed characters. <CR> = 13 (Dec) and <LF> = 10 (Dec).
RULE 5 :	The data file must end with an END OF FILE character. <EOF> = 26 (Dec).

Following these rules is especially easy when programming in BASIC. BASIC's built-in WRITE# function automatically adheres to rules 2, 4, and 5. Therefore, <CR>, <LF>, and <EOF> characters are automatically inserted.

Functions of Program Lines in Listing 1

70	Merges the PCI Header files containing the definition and location of the drivers.
1030 to 1070	Variable declarations. These parameters can be altered to suit unique requirements.
1110 to 1120	Defines array sizes.
1160 to 1170	Initializes the PCI-20000 software and hardware.
1210 to 1260	Acquires the data and stores it in arrays.

1300	Opens the disk file (it must have a .PRN extension to be read by Lotus).
1340 to 1420	Shows how to write the PCI-20000 data to the file.
1400	Converts the raw A/D data to volts.
1410	Writing data. Notice the comma separators for multiple-row data.
1460	Closing the file.

To use this program simply LOAD it, define the variables in lines 1030 to 1070, connect an input to the channel defined by CHANNO, and RUN it. The program will create a file named FILENAME$, that can be read by Lotus. Once data is inside the Lotus environment, all of the features and power of this product are available.

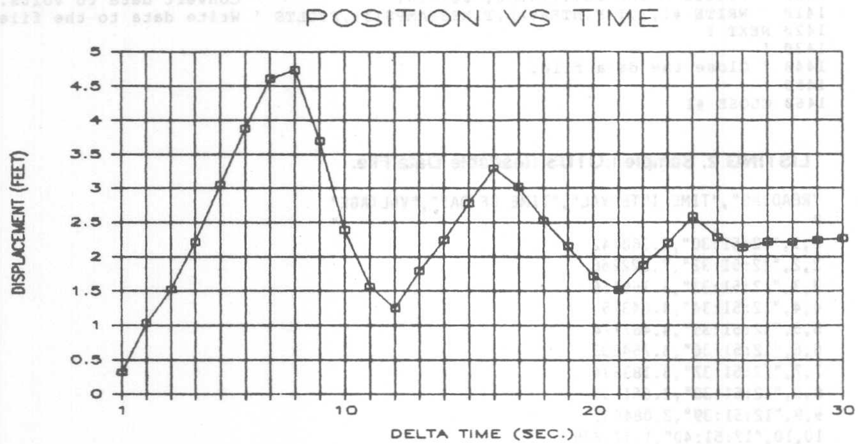


Figure 1. An example of a LOTUS 1-2-3 Graph

LISTING 1. A BASIC Program to Acquire Analog Data and Write it to a LOTUS Readable Disk File.

```
10    ' Sample Program to Demonstrate:  Interfacing Data-Files to
20    ' LOTUS 1-2-3, Writing files to Disk & Generating Time-bases.
30    '
40    ' The System:  PCI-20002M-1  A/D Module (+/- 10 V range)
50    ' ----------   PCI-20046S-1  Software Drivers
60    '
70    CHAIN MERGE "PCIHEAD.BAS", 100, ALL
80    '
1000  ' Define program variables.
1010  ' The following terms can be changed to suit requirements.
1020  '
1030  FILENAME$ = "A:SAMPLES.PRN"    ' Name of file to receive data.
1040  PASSES    = 30                 ' Number of data points to take.
1050  CARSEG    = &HD000             ' Base address of carrier in RAM.
1060  CHANNO    = 0                  ' Channel number to read.
1070  FINTERVAL = 1!                 ' Interval between readings in sec.
1080  '
1090  ' Declare array sizes.
1100  '
1110  DIM  VALUES(PASSES)
1120  DIM  TIMESTAMP$(PASSES)
1130  '
1140  ' Initialize the PCI-20000 Hardware and Software
1150  '
1160  CALL SYSINIT
1170  CALL INIT(CARSEG)
1180  '
1190  ' Acquire the data (Analog & Time) and save in arrays.
1200  '
1210  FOR I = 1 TO PASSES
1220    FTIME = TIMER                              ' Initialize the timer.
1230    IF (FTIME+FINTERVAL) > TIMER THEN GOTO 1230 ' Delay for interval.
1240    CALL READ.CH(AI, CHANNO, VALUES(I))        ' Read the analog input.
1250    TIMESTAMP$(I) = TIME$                      ' Get and save time stamp.
```

```
1260 NEXT I
1270 '
1280 ' Open the data file.
1290 '
1300 OPEN FILENAME$ FOR OUTPUT AS #1
1310 '
1320 ' Write header information to the file.
1330 '
1340 WRITE #1,"READING","TIME INTERVAL","TIME OF DAY","VOLTAGE"
                                                          "
1350 WRITE #1,"
1360 '
1370 ' Write the data form the arrays to the disk file.
1380 '
1390 FOR I = 1 TO PASSES
1400   FVOLTS = VALUES(I)/4096!*20!-10!              ' Convert data to volts.
1410   WRITE #1,I,I*FINTERVAL,TIMESTAMP$(I),FVOLTS ' Write data to the file.
1420 NEXT I
1430 '
1440 ' Close the data file.
1450 '
1460 CLOSE #1
```

LISTING 2. Sample LOTUS Readable Data File.

```
"READING","TIME INTERVAL","TIME OF DAY","VOLTAGE"
"                                               "
1,1,"12:51:30",2.260742
2,2,"12:51:32",2.822266
3,3,"12:51:33",3.759766
4,4,"12:51:34",4.84375
5,5,"12:51:35",4.467774
6,6,"12:51:36",3.754883
7,7,"12:51:37",3.183594
8,8,"12:51:38",2.661133
9,9,"12:51:39",2.084961
10,10,"12:51:40",1.171875
11,11,"12:51:41",.5322266
12,12,"12:51:42",0
13,13,"12:51:43",.4394532
14,14,"12:51:44",1.230469
15,15,"12:51:45",2.050781
16,16,"12:51:46",2.851563
17,17,"12:51:47",2.939453
18,18,"12:51:48",2.431641
19,19,"12:51:49",1.99707
20,20,"12:51:50",1.367188
21,21,"12:51:51",1.21582
22,22,"12:51:52",1.811524
23,23,"12:51:53",2.407227
24,24,"12:51:54",2.993164
25,25,"12:51:56",3.15918
26,26,"12:51:57",2.700195
27,27,"12:51:58",2.25586
28,28,"12:51:59",1.938477
29,29,"12:52:00",2.15332
30,30,"12:52:01",2.636719
```

GRAPHING WITH LOTUS 1-2-3

There are often instances when it is desirable to produce a graphic display of recorded data. This can have many uses including notebook entries, reports and presentations. Most data acquisition systems do not inherently have plotting capabilities. However, most can be interfaced to appropriate software which can perform this function. Some products, such as the ControLOGraph, can generate useful graphs directly. In specific applications when detailed control over plot format, size, labeling or other characteristics are desired, it is sometimes helpful to call upon the extensive plotting features of Lotus 1-2-3.

This applications note assumes that the reader has a basic working knowledge of 1-2-3. Presented here is a menu-driven 'macro' (once installed) that reads a given data file and plots the results. The details of the plot characteristics are easily customized by the user. Naturally, the input data file must be in an appropriate Lotus-readable format. The requirements for such a file are defined in the preceding applications note, *Programming Your PC for Data Acquisition Applications.* File conversion techniques are also described.

The following steps are required to use the macro:

- Enter the macro as shown in listing 1. Be sure to start with the first entry, "\G", in cell AA1
- Name the macro as follows: /RNC\G <enter> AB6 <enter>
- To run the macro, type: Alt-G

The user will be prompted for the name of the data file to be plotted and the graph title.

Listing 2 shows a simple example of a data file as read from the PCI ControLOGraph. Note that the ControLOGraph automatically generates the output file in the correct format, so that no conversion is required. Figure 1 suggests just one of the possible ways that this data could be presented.

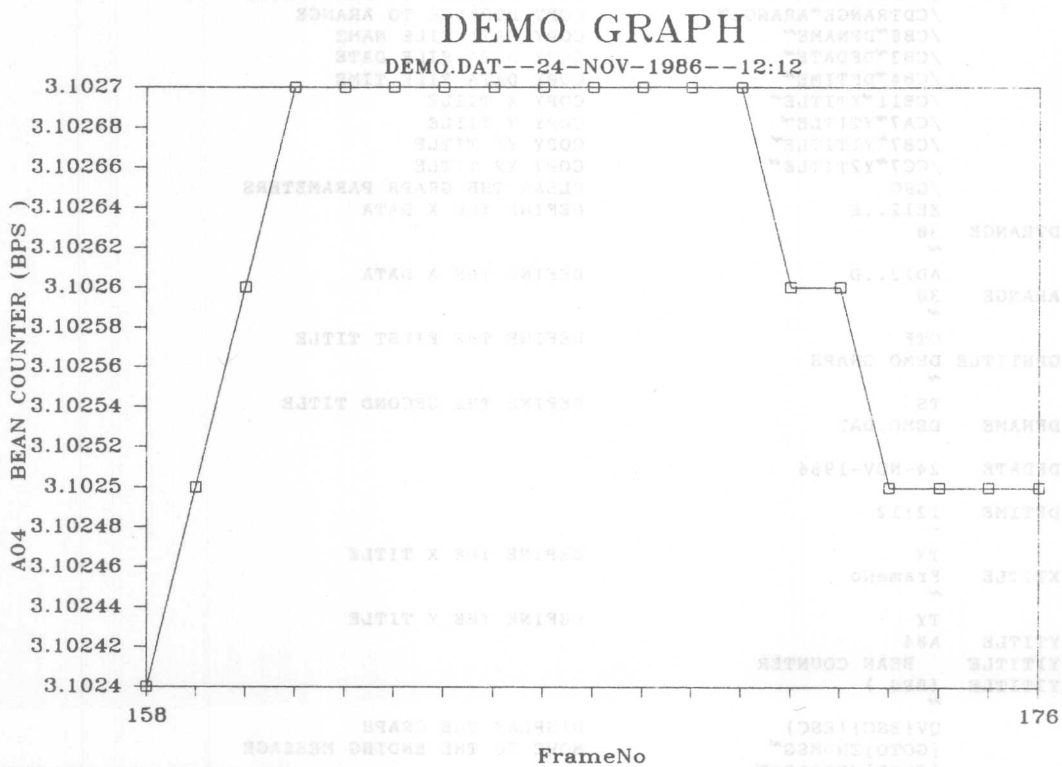


FIGURE 1. A Sample Lotus Plot.

Listing #1.

```
\G        THIS MACRO AUTOMATICALLY PRODUCES A GRAPH FROM ControLOGraph DATA

LSTBLROW        29
LSTDTROW        30
BLNKCELL
          {HOME}                        MOVE TO CELL A1
          /RNLRAA3..AA140~              NAME RANGES TO BE USED
          /RNCFNAME~AE81~               CREATE FNAME
          /RNCGTITLE~AF100~             CREATE GTITLE
          /RNCMESSAGE~AE123~            CREATE MESSAGE
          {GOTO}FILENAME~               MOVE TO THE FILENAME PROMPT MENU
          {GOTO}FNAME~                  MOVE TO THE FNAME FIELD
          {?}~                          GET THE FILENAME
          /CFNAME~GETFILE~              COPY THE ENTERED FILENAME
          {GOTO}GRPHTITLE~              MOVE TO THE GRAPH TITLE PROMPT MENU
          {GOTO}GTITLE~                 MOVE TO THE GTITLE FIELD
          {?}~                          GET THE TITLE
          /CGTITLE~GPHTITLE~            COPY THE ENTERED TITLE
          {HOME}                        MOVE TO CELL A1
          /FIN                          READ IN THE DATA
GETFILE   DEMO
          ~
          /CLSTBLROW~BLRANGE~           COPY LSTBLROW TO BLRANGE
          {GOTO}BLRANGE~                MOVE CURSOR TO BLRANGE
          {EDIT}{CALC}{HOME}'~          MAKE THE VALUE A STRING
          /CBLNKCELL~E13..E             COPY BLANKS INTO THE RANGE
BLRANGE   29
          ~
          /CLSTDTROW~DTRANGE~           COPY LSTDTROW TO DTRANGE
          {GOTO}DTRANGE~                MOVE CURSOR TO DTRANGE
          {EDIT}{CALC}{HOME}'~          MAKE THE VALUE A STRING
          {GOTO}FILENAME~               MOVE CURSOR TO FILENAME
          {GOTO}FNAME~                  MOVE CURSOR TO FNAME FIELD
          /CDTRANGE~ARANGE~             COPY DTRANGE TO ARANGE
          /CB8~DFNAME~                  COPY DATA FILE NAME
          /CB3~DFDATE~                  COPY DATA FILE DATE
          /CB4~DFTIME~                  COPY DATA FILE TIME
          /CE11~XTITLE~                 COPY X TITLE
          /CA7~YTITLE~                  COPY Y TITLE
          /CB7~Y1TITLE~                 COPY Y1 TITLE
          /CC7~Y2TITLE~                 COPY Y2 TITLE
          /GRG                          CLEAR THE GRAPH PARAMETERS
          XE12..E                       DEFINE THE X DATA
DTRANGE   30
          ~
          AD12..D                       DEFINE THE A DATA
ARANGE    30
          ~
          OTF                           DEFINE THE FIRST TITLE
GPHTITLE  DEMO GRAPH
          ~
          TS                            DEFINE THE SECOND TITLE
DFNAME    DEMO.DAT

DFDATE    24-NOV-1986

DFTIME    12:12
          ~
          TX                            DEFINE THE X TITLE
XTITLE    FrameNo
          ~
          TY                            DEFINE THE Y TITLE
YTITLE    A04
Y1TITLE     BEAN COUNTER
Y2TITLE   (BPS )
          ~
          QV{ESC}{ESC}                  DISPLAY THE GRAPH
          {GOTO}ENDMSG~                 MOVE TO THE ENDING MESSAGE
          {GOTO}MESSAGE~
```

Continuation of Listing #1.

```
FILENAME +------------------------------------------------------------------+
         |                                                                  |
         |          ENTER THE ControLOGraph .PRN FILE TO BE GRAPHED         |
         |                                                                  |
         |                              DEMO                                |
         |                                                                  |
         |                                                                  |
         +------------------------------------------------------------------+

GRPHTITLE+------------------------------------------------------------------+
         |                                                                  |
         |          ENTER THE GRAPH TITLE :   DEMO GRAPH                    |
         |                                                                  |
         |                                                                  |
         |                                                                  |
         +------------------------------------------------------------------+

ENDMSG   +------------------------------------------------------------------+
         |                                                                  |
         |                    ALT-G EXECUTES THE MACRO AGAIN                |
         |                                                                  |
         |                                                                  |
         |                                                                  |
         +------------------------------------------------------------------+
```

Note: Figures not shown actual size. Create box to fill entire screen.

Listing #2.

```
"AC"
"PCI ControLOGraph"
"ASCII File Date = ","24-NOV-1986"
"ASCII File Time = ","12:12"
"Trigger Frame No = ",13
"Origin Frame No = ",1
"A04  ","  BEAN COUNTER ","(BPS )"
"Acquisition Data File = ","DEMO.DAT"
"Total Points Written = ",44
"Trigger Relative Time"
"Date","Time","tX Values","tY Values","FrameNo"
"31-DEC-1985","23:55:28.4"," 0000:36:15.0", 3.1024E+00,158
"31-DEC-1985","23:55:43.4"," 0000:36:30.0", 3.1025E+00,159
"31-DEC-1985","23:55:58.4"," 0000:36:45.0", 3.1026E+00,160
"31-DEC-1985","23:56:13.4"," 0000:37:00.0", 3.1027E+00,161
"31-DEC-1985","23:56:28.4"," 0000:37:15.0", 3.1027E+00,162
"31-DEC-1985","23:56:43.4"," 0000:37:30.0", 3.1027E+00,163
"31-DEC-1985","23:56:58.4"," 0000:37:45.0", 3.1027E+00,164
"31-DEC-1985","23:57:13.4"," 0000:38:00.0", 3.1027E+00,165
"31-DEC-1985","23:57:28.4"," 0000:38:15.0", 3.1027E+00,166
"31-DEC-1985","23:57:43.4"," 0000:38:30.0", 3.1027E+00,167
"31-DEC-1985","23:57:58.4"," 0000:38:45.0", 3.1027E+00,168
"31-DEC-1985","23:58:13.4"," 0000:39:00.0", 3.1027E+00,169
"31-DEC-1985","23:58:28.4"," 0000:39:15.0", 3.1027E+00,170
"31-DEC-1985","23:58:43.4"," 0000:39:30.0", 3.1026E+00,171
"31-DEC-1985","23:58:58.4"," 0000:39:45.0", 3.1026E+00,172
"31-DEC-1985","23:59:13.4"," 0000:40:00.0", 3.1025E+00,173
"31-DEC-1985","23:59:28.4"," 0000:40:15.0", 3.1025E+00,174
"31-DEC-1985","23:59:43.4"," 0000:40:30.0", 3.1025E+00,175
"31-DEC-1985","23:59:58.4"," 0000:40:45.0", 3.1025E+00,176
"01-JAN-1986","00:00:13.4"," 0000:40:59.9", 3.1024E+00,177
"01-JAN-1986","00:00:28.4"," 0000:41:14.9", 3.1024E+00,178
"01-JAN-1986","00:00:43.4"," 0000:41:29.9", 3.1024E+00,179
"01-JAN-1986","00:00:58.4"," 0000:41:44.9", 3.1024E+00,180
```

PEDAS

A Personal Expert Data Acquisition System

ABSTRACT

Many industrial and laboratory tasks typically require an individual with expertise in specific methods of problem recognition and resolution. This paper shows how a personal computer, data acquisition system and expert system software can be combined to create a low-cost experimental personal expert data acquisition system which allows the knowledge of an expert to be used by nonexperts for laboratory and industrial instrumentation, test, measurement and control applications.

INTRODUCTION

An expert system is a computer program that aids in solving complex "real-world" problems usually requiring a human expert. Expert system software allows a human "expert" to outline the process used to solve a problem. Once this has been done, the outline can be used to generate a series of questions that lead a nonexpert through the solution of similar problems. Although a human expert can never be entirely replaced, access to his or her expertise can be extended.

The outline used to solve a problem is represented in the form of rules. A rule is a fact which states that IF a certain set of statements is true, THEN the final statements must also be true. During a consultation, the expert system uses what is called an "inference engine" to choose which rules to try to prove. Each fact in a rule can be proven either by using another rule that draws conclusions about that fact or by asking the user for information. The rules chosen will change depending on the answers given during a consultation. The set of rules used to solve a particular problem are contained in what is called a knowledge base.

There are many industrial and laboratory data acquisition, test, measurement and diagnosis tasks that have typically required an individual with expertise in specific methods of problem recognition and resolution. For example, in diagnosing equipment failures, a technician typcially draws conclusions as to specific faults based on his personal interpretation of the data he has acquired. The problem with this method is that the speed of repair, and therefore the cost, is based on the skills and experience a particular individual has with a particular problem. When performing repairs, the time spent diagnosing the problem is not particularly productive. Rather, the objective is to fix the problem correctly.

To minimize the expertise required and time spent diagnosing a problem, many automobile manufacturers are using expert systems for vehicle fault diagnosis. A data acquisition interface system allows a computer to obtain information directly from the vehicle being tested. The computer software analyzes the acquired data, as well as answers to questions presented to the operator, and presents a conclusion to the repair technician. The expert system reaches the same conclusion as an 'expert' technician would given the same inputs. The computer software, however, was developed using the inputs from a number of expert technicians and engineers. In addition, the computer usually operates faster than most experts and is consistent, always drawing the same conclusions from the same set of data.

Expert systems do have disadvantages, however. Often there are intangibles that a human expert recognizes but a computer cannot. Also, it may not be economically possible to provide all the information necessary to thoroughly diagnose a particular problem. Furthermore, the expert system may not always be able to diagnose the problem to the least expensive solution.

A GENERAL-PURPOSE PERSONAL EXPERT DATA ACQUISITION SYSTEM

The main objective when creating PEDAS, an experimental personal expert data acquisition system, was to create a system that could be used for 'real-world' scientific and industrial applications. This required the expert system software to test rules directly by taking appropriate measurements or readings using a data acquisition system. The operator is still prompted for inputs when the expert system cannot directly obtain an answer. To minimize cost, PEDAS uses an inexpensive personal computer, a configurable board-level data acquisition system and popular expert software. The system described in this paper can be assembled for about $3,000 (US dollars).

THE PERSONAL COMPUTER

The decision to use a personal computer for PEDAS was easily made. The economical and powerful personal computer has allowed more people to work on more applications for more computers than ever before in history. By 1986, there were over 400,000 PC's being used in scientific and industrial applications world-wide. What makes the personal computer so attractive to so many scientists and engineers? The single most important feature is price. The introduction of the IBM PC personal computer in 1981 created a de facto standard for others to follow. One after another, computer manufacturers started copying—and improving—the IBM PC design. Mass production and mass marketing made the PC what it is today—powerful computing at low cost. Computers can now be applied in hundreds of factory and laboratory applications that previously were not economically possible. Complete IBM PC-compatible personal computer systems are currently available in the US market for under $1,500.

THE EXPERT SYSTEM SOFTWARE

To allow PEDAS to obtain data directly from the "real-world", the expert system software must be extensible to allow for the integration of the data acquisition support software. It must also be inexpensive and run on an IBM PC-compatible computer. The expert system software chosen for PEDAS was the MicroExpert, available from McGraw-Hill Book Company for about $50 (US dollars).

MicroExpert is a PC-based expert system written in Turbo Pascal. It comes complete with source code and an extensive manual explaining its design, use, and extension. MicroExpert has all the components required to make it a useful tool for the development of application-specific expert systems. Among these are a rule parser for creating the knowledge base, a prompt parser for creating the user interface, a consultation system for executing the knowledge base, translations for enhancing the user interface, and the source code for customization.

THE DATA ACQUISITION SYSTEM

Flexible and cost-effective modular data acquisition systems that plug directly into a personal computer have become quite popular for many scientific and industrial applications. These modular systems separate the circuitry required by the personal computer from the requirements of the application. A 'mother' board, which serves the data format and timing requirements of the PC, carries function-dependent 'daughter-boards.' With features like analog input, digital input and output, pulse counting, analog output and others, daughter-boards can be combined to handle the requirements of a particular application. The entire mother-daughter combination plugs directly into a PC expansion slot. The system chosen for PEDAS in the PCI-20000 Personal Computer Interface System, which can support from one to 128 digital input/outputs, one to 80 single-ended analog inputs, or one to 24 analog outputs on a single mother board. Many other interface combinations are possible with this system. The specific data acquisition system components used in PEDAS include the PCI-20001C-2 Carrier with digital I/O, the PCI-20002M-1 Data Acquisition module and the PCI-20046S-3 Pascal Software Support package.

THE HARDWARE/SOFTWARE INTERFACE

The PCI-20000 data acquisition system is supported by a set of software drivers that isolate the programmer from the details of the hardware. These drivers are callable from several high-level languages including Turbo Pascal. Since MicroExpert is also written in Turbo Pascal and allows user-written functions and procedures, it was easily modified to access the data acquisition system. The resulting PEDAS software supports hardware configuration, I/O read and write, and analog input acquisition. These modifications are outlined below.

1. In the MICROEXP.PAS file, insert the following directly after "Utility Routines":

```
{$I PCIHEAD.PAS}
{$I P46. PAS    }
```

2. In the MICRO2.PAS file, insert the following directly after the line "Put user functions and procedures here.":

```
Function DITest(Parameter : Parm_Array) : Boolean;
Var
   Channel, TestValue, Reading : Integer;
Begin { DITest }
   Channel      := ToInteger(Parameter[1]) ;
   Test Value := ToInteger(Parameter[2]) ;
   Reading      ;= ReadCH(_DIBT, Channel)  ;
   DI Test      ;= FALSE ;
   If Reading = TestValue Then
   Begin
      DITest  := TRUE ;
   End ;
End ; { DITest }

Function AITest(Parameter : Parm_Array) : Boolean;
Var
   Channel    , AIReading ,
   UpperBound, LowerBound : Integer ;
   Flag                   : Boolean ;
Begin { AITest }
   Channel      := ToInteger(Parameter[1]) ;
   UpperBound := ToInteger(Parameter[2]) ;
   LowerBound := ToInteger(Parameter[3]) ;
   AIReading  := ReadCh(_AI, Channel)     ;
   Flag := False;
   If (AIReading < LowerBound) or
      (AIReading > UpperBound) Then
   Begin
      Flag:=True;
   End;
   If Parameter[4] = 'INSIDE' Then
   Begin
      Flag := Not Flag '
   End;
   AITest := Flag ;
End ; { AITest }

Procedure WriteDO(Parameter :Parm_Array);
Var
   Channel, Value : Integer ;
Begin { WriteDO }
   Channel := ToInteger(Parameter[1]) ;
   Value   := ToInteger(Parameter[2]) ;
   WriteCh(_DOBT, Channel, Value) ;
End ; { WriteDO }

Function ConfigAI(Parameter : Parm_Array) : Boolean ;
Var
   Channel, GainValue : Integer ;
Begin { ConfigAI }
   Channel    := ToInteger(Parameter[1]) ;
   GainValue := ToInteger(Parameter[2]) ;
   CnfAI(Channel, GainValue, -1, 1) ;
   ConfigAI  := FALSE;
   If ErrSys = 0 then
   Begin
        ConfigAI := TRUE;
   End;
End ; { ConfigAI }

Function Init20K : Boolean ;
Begin { Init20K }
   Init20K := False ;
   SetVec ($60)     ;
   SysInit ;
   Init ($C000) ;
   Iff ErrSys = 0 Then
   Init20K := True ;
   End ;
End ; { Init20K }

Function ConfigDO(Parameter : Parm_Array) : Boolean ;
Var
   Channel, InitialValue : Integer ;
Begin { ConfigDO }
   Channel          := ToInteger(Parameter [1]) ;
   Initial Value := ToInteger(Parameter [2]) ;
   CnfDO(Channel, Initial Value) ;
   ConfigDO := False ;
   If ErrSys = 0 Then
```

9

APPLICATIONS, TOOLS & TECHNIQUES
EXPERT SYSTEMS

```
   Begin
      ConfigDO := True ;
   End ;
End ; { ConfigDO }

Function ConfigDI(Parameter : Parm_Array) : Boolean ;
Var
   Channel : Integer ;
Begin { ConfigDI }
   Channel := ToInteger(Parameter [1]) ;
   CnfDI(Channel) ;
   ConfigDI := False ;
   If ErrSys = 0 Then
  Begin
      ConfigDI := True ;
   End;
End ; { ConfigDI }

Function SysError : Boolean ;
Begin { SysError }
   SysError := True;
   If ErrSys = 0 Then
   Begin
      SysError := False ;
   End;
End ; { SysError }

Procedure DispMsg(Parameter :Parm_Array);
Var
   Index : Integer;
Begin { DispMsg }
   Index := 1 ;
   While Parameter [Index] <> '' Do
   Begin
      Write(Parameter [Index]);
      Index := Index + 1        ;
   End;
   WriteLn ;
End ; { DispMsg }
```

3. Still in "MICRO2.PAS", find "PROCEDURE evalue_func ;". Insert the following after the line that states "* Insert calls to user functions here. *":

```
Else If f_name = 'DITEST'
   Then Func_Value := DITest(Parm)
Else If f_name = 'AITEST'
   Then Func_Value := AITest(Parm)
Else If f_name = 'CONAI'
   Then Func_Value := ConfigAI(Parm)
Else If f_name = 'SYSINIT'
   Then Func_Value := Init20K
Else If f_name = 'CONDO'
   Then Func_Value := ConfigDO(Parm)
Else If f_name = 'CONDI'
   Then Func_Value := ConfigDI(Parm)
Else If f_name = 'ERRSYS'
   Then Func_Value := SysError
```

4. In "MICRO2.PAS" after the line "THEN math (parm) (* insert calls to new procedures here *)" insert the following:

```
Else If f_name = 'WRITEDO'
   Then WriteDO(Parm)
Else If f_name = 'DISPMSG'
   Then DispMsg(Parm)
```

5. The last step is creating the Knowledge Base for your specific application. The specific syntax for creating rules, prompts and translations are outlined in the MicroExpert manual. The syntax for the functions described above, analog input and digital input/output, follows.

```
FUNCTION CONAI ('channel', 'gain')
   This function configures the selected analog
   input channel to a gain of 1, 10, 100, or 1000.
Example:  FUNCTION CONAI ('0','10')
   configures analog input 0 to a gain of 10.

FUNCTION CONDO ('channel','value')
   This function configures a digital group
   containing the selected 'channel' to outputs
   with an initial group state of 'value.'
Example:  FUNCTION CONDO ('0', '0')
   configures the digital group containing channel
   0 to outputs with all eight bits set low.

FUNCTION CONDI ('channel')
   This function configures a digital group
   containing the selected 'channel' to inputs.
Example:  FUNCTION CONDI ('8')
   configures the digital group containing channel
   8 as inputs.

FUNCTION SYSINIT
   This function initializes the support software
   system with the PCI-20000 carrier at address
   C000 (hex). TRUE is returned if the system is
   configured properly.

FUNCTION DITEST ('channel','value')
   This function tests the digital inputs 'channel'
   to see if it is equal to 'value.'
Example:  FUNCTION DITEST ('8','1')
   tests digital input channel 8 and returns TRUE
   if it is 1, FALSE if it is 0.

FUNCTION AITEST ('channel', 'upper-bound',
   'lower-bound', 'sense')
   • If the analog input 'channel' is between 'upper-
   bound' and 'lower-bound' a TRUE is returned if
   'sense' is 'INSIDE'.
   • If the analog input 'channel' is not between
   'upper-bound' and 'lower-bound' a FALSE is
   returned if 'sense' is 'INSIDE.'
   • If the analog input 'channel' is between 'upper-
   bound' and 'lower-bound' a FALSE is returned if
   'sense' is 'INSIDE.'
   • If the analog input 'channel' is between 'upper-
   bound' and 'lower-bound' a FALSE is returned if
   'sense' is 'OUTSIDE.'
   • If the analog input 'channel' is not between
   'upper-bound' and 'lower-bound' a TRUE is
   returned if 'sense' is 'OUTSIDE.'
Example:  FUNCTION AITEST ('0', '3', '1', 'OUTSIDE')
   returns a TRUE for an analog input value of 4,
   a FALSE for an analog input value of 2.

FUNCTION ERRSYS
   This function returns TRUE if there is an error
   generated by the data acquisition software support.

PROCEDURE WRITEDO ('channel', 'value')
   This procedure sets the digital output 'channel'
   to a 'value' of 0 or 1.
Example:  PROCEDURE WRITEDO ('3', '1')
   sets digital output channel 3 to a value of 1.

PROCEDURE DISPMSG ('word 1', ... 'word n')
   This procedure displays to words 'word 1' through
   'word n' on the display followed by a carriage
   return and line feed. The maximum value for n is 10.
Example:  PROCEDURE DSPMSG ('Pass')
```

With the above changes and the instructions in the MicroExpert and PCI-20000 manuals, expert system rules can be designed that directly use "real-world" data. These rules can be used in creating knowledge bases for specific scientific and industrial applications.

SUMMARY

A personal computer, expert system software and a plug-in data acquisition system can be easily combined to create the low-cost experimental Personal Expert Data Acquisition System (PEDAS). PEDAS can be used for equipment fault diagnosis and repair, data acquisition and interpretation, failure analysis, product testing and other laboratory and industrial instrumentation, test, measurement and control applications.

PCI-20000

THE NEW GENERATION OF PERSONAL COMPUTER INSTRUMENTATION

TABLE OF CONTENTS

Free Burr-Brown Demonstration Diskettes showing product capabilities, specifications, and applications for the PCI-20000 system are available through Burr-Brown sales offices. These diskettes run on the IBM PC and compatible computers containing a graphics card. Please contact your local sales office for your free diskette. See office listings at the back of this handbook.

PCI-20000 SYSTEM

A New Concept In Flexibility And Cost Efficiency

PCI-20000 is an all-new, high technology, computer-based instrumentation system which allows a user to specify *exactly* the system needed for his particular application, and to buy only that system—no more, no less—without giving up extensive expansion capabilities for the future.

Modularity–The Key Concept For Optimum System Design

The key concept of the PCI-20000 is modularity. The PCI-20000 system is configured using two very basic types of printed-circuit boards: **Carriers plus Instrument Modules.**

Modular PCI-20000.

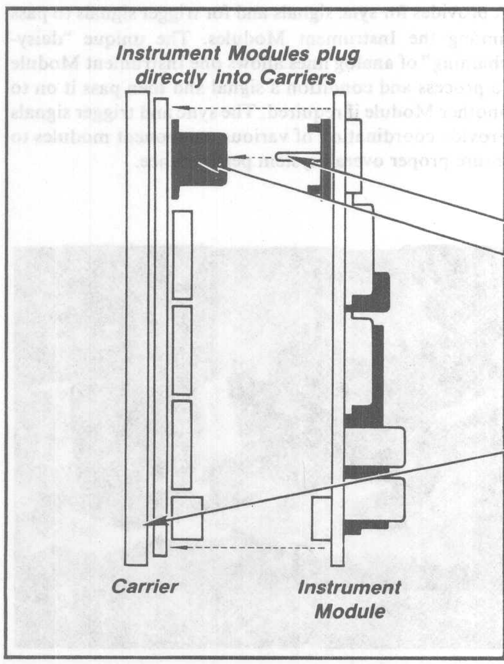


Illustration of the Modular Approach to Personal Computer Instrumentation.

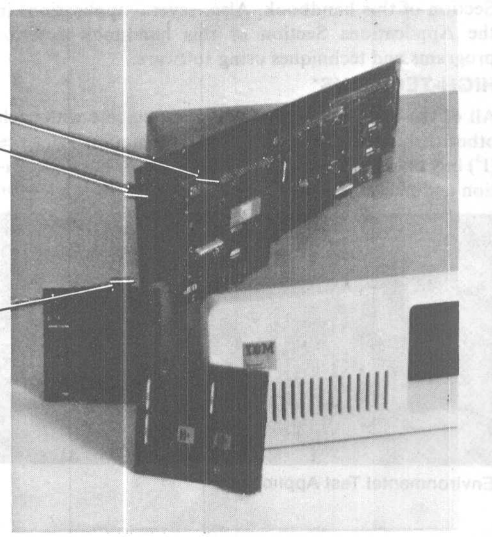
The Instrument Modules are designed to interface with and to physically plug into various Carriers. The "mother-board" Carrier interfaces with the host computer. For example, Carriers plug directly into expansion slots of personal computers such as the IBM PC, XT, AT, and other compatible personal computers and handle all interfacing of the computer bus. Carriers normally provide power, inter-module communications, and physical mounting mechanisms for the instrument modules. Optional Carriers also provide 32 points of fully buffered, TTL-compatible, digital I/O self-contained on the Carrier itself.

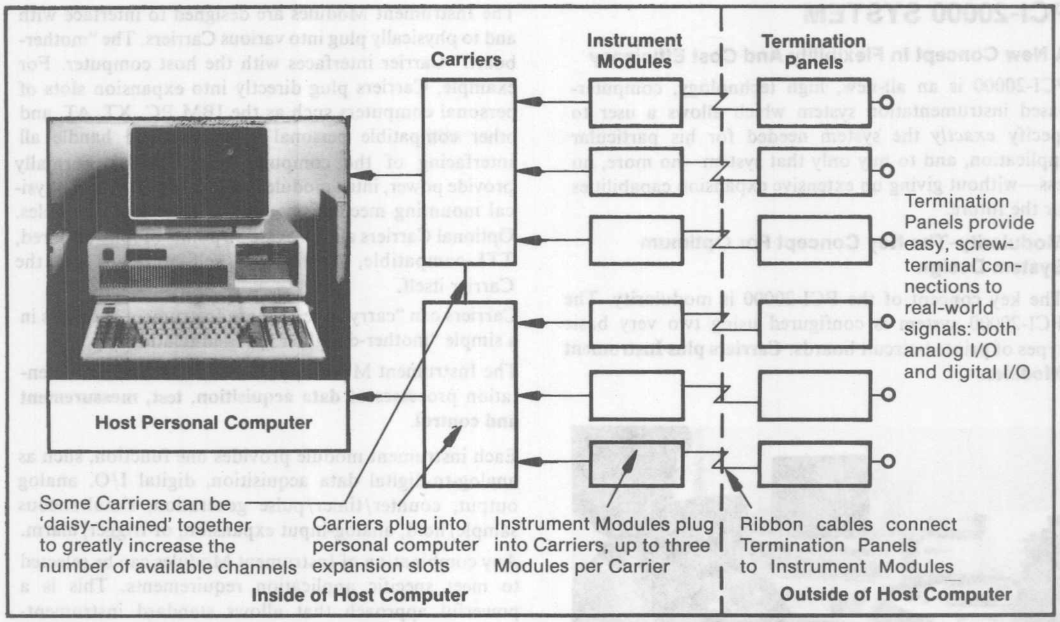
Carriers can "carry" one or more Instrument Modules in a simple "mother-daughter" plug-in relationship.

The Instrument Modules perform the actual instrumentation processes of **data acquisition, test, measurement and control**.

Each instrument module provides one function, such as analog-to-digital data acquisition, digital I/O, analog output, counter/timer/pulse generation, simultaneous sample/hold, analog-input expansion, or trigger/alarm. Any combination of Instrument Modules can be selected to meet specific application requirements. This is a powerful approach that allows standard instrument-modue components to be specified and configured by the user into a system optimized for his individual requirements. This maximizes performance and minimizes cost, since the user needs to buy only those modules required to get his job done satisfactorily.

As requirements change, different Instrument Modules can be substituted to solve a new problem. More Instrument Modules can be added if requirements expand. As new, improved modules are added to the product line, performance can be upgraded easily.





PCI-20000 System Illustration.

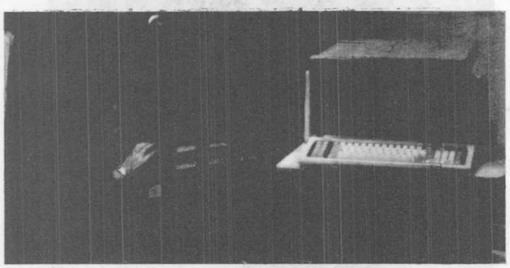
OUTSTANDING SOFTWARE SUPPORT

A wide range of software packages are available for the PCI-20000. There are general-purpose software packages that support all instrument modules and offer a choice of languages. All general-purpose software supports thermocouples at no extra cost to the user. There are additional high-performance software packages available that perform more specialized functions. Various packages are summarized later in the Specification Summary. For more complete information, please refer to the Software Section of this handbook. Also, several applications in the Applications Section of this handbook describe programs and techniques using software.

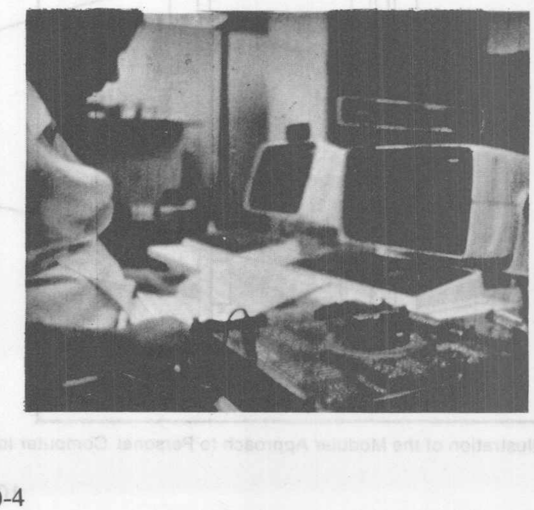
HIGH-TECH BUS*

All of the Instrument Modules communicate with each other through an Intelligent Instrumentation Interface (I^3) bus on the Carrier that is optimized for data acquisition and measurement.

*Patent applied for.


Environmental Test Application.

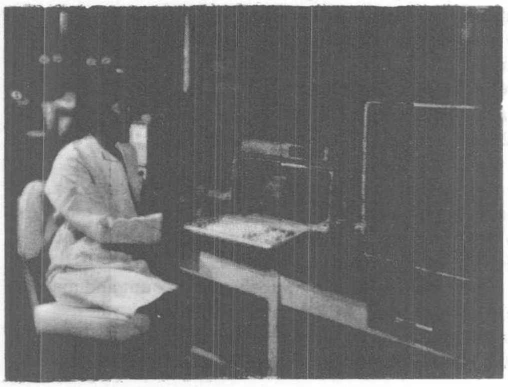
This bus allows the standard types of memory and I/O access to occur in the fashion normally found on a computer bus. The I^3 bus also provides for the "chaining" of analog signals among the plug-in Instrument Modules. It provides for sync signals and for trigger signals to pass among the Instrument Modules. The unique "daisy-chaining" of analog lines allows one Instrument Module to process and condition a signal and then pass it on to another Module if required. The sync and trigger signals provide coordination of various component modules to insure proper overall system performance.



PCI-20000 MAKES OBSOLESCENCE OBSOLETE

The thought put into this I^3 bus permits new Instrument Modules with more features or higher performance to be added to a system as technology improves, without affecting the Modules configured initially. This insures against system obsolescence, since Modules for *critical functions* can be upgraded to achieve higher performance without the necessity of replacing a whole system at that time.

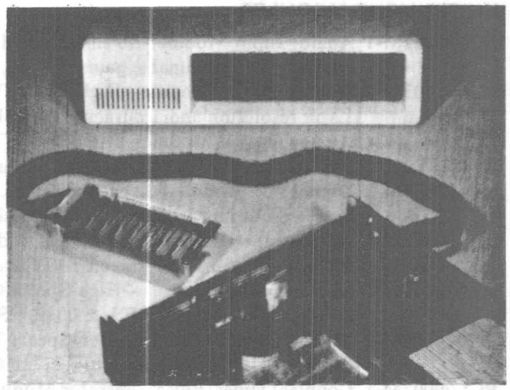
Also, a system can be expanded, if the number of channels or types of measurements or controls required increase, by adding Instrument Modules to vacant positions on a Carrier, and by adding additional Carriers to accommodate the new instrument components as required.



Laboratory Application.

The classic deficiencies found in other single-board data-acquisition systems are largely eliminated in this design.

1. A wide variety of I/O types are supported: analog input (voltage, current, thermocouple, etc . . .), simultaneous multiple channel readings, analog output, digital input, digital output, counter input, frequency measurement, pulse generation along with high speed triggering and alarm monitoring.
2. Relatively large numbers of channels can be accommodated. Up to 128 digital I/O points or 80 analog inputs or 24 analog outputs or 12 counter/timer ports can be configured on a single carrier, using the instrument modules available. In addition, multiple carriers can be used simultaneously.
3. By selecting the appropriate mix of modules, the capability of the system can be tailored to a particular application. Thus, function and cost can be optimized. Modules can be added, changed or rearranged at any time to satisfy new measurement or control requirements.
4. Probable conflicts in available address space within the PC have been eliminated by the choice of memory versus I/O mapping. Only 2K bytes are decoded on the I/O side of the IBM PC, while hundreds of kilobytes are potentially available in RAM address space. Switches on the carrier allow placing the unit anywhere in the PC's memory map.
5. In addition to the memory and digital I/O connections provided by a standard computer bus, the internal Intelligent Instrumentation Interface (I^3) bus also provides for analog, synchronization and trigger signal routing. This facility allows the chaining of analog signals from one module to another as well as the triggering or sychronizing of events on a particular module by another module.



Shielded Cable Connects Instrument Module Inside the PC to an Outside Termination Panel.

Some of the data acquisition systems on the market today do not clearly state how fast they can acquire data. Performance can be measured in many ways, some more meaningful than others. For example, on the subject of data acquisition speed, "Conversion Time" is often quoted. However, as was described in the earlier portions of this handbook, the A/D's Conversion Time is but one of many important components of speed. When selecting a DA&C system it is necessary to know not only the "real" specifications but under what conditions they apply.

The SPEED SUMMARY TABLE and the individual PRODUCT DATA SHEETS found later in this section and in the next section of this handbook, clearly define the detailed performance characteristics of the PCI-20000 system. An overview of the capabilities are presented, here, in the SPECIFICATION SUMMARY TABLE. Please note that under most conditions the PCI-20000 is faster, without DMA, than most other systems are using DMA. The unique features of PCI DMA not only include very high speed (360K bytes/Sec to RAM) but it is also the only system that can transfer Analog I/O, Digital I/O and Counter/Frequency data.

Hundreds of different system configurations can be assembled by intermixing the wide variety of components within the PCI-20000 family. This permits each unique application to be satisfied without cost and performance compromises. A number of TYPICAL CONFIGURATIONS is outlined later in this section and suggests a variety of useful combinations.

TABLE 10.1. PCI-20000 SPECIFICATION SUMMARY

CARRIERS

PCI-20001C-1	**Carrier Board**—Plugs into an expansion slot of any IBM PC, XT, AT or other compatible computers. Holds up to three Instrument Modules.
PCI-20001C-2	**Carrier Board**—Same as PCI-20001C-1 with the addition of 32 points of digital I/O.
PCI-20041C-2	**High Performance Carrier Board**—Same as PCI-20001C-2 with the addition of an inter-carrier bus, a programmable Pacer/Timer and external Interrupt/Acquisition control.
PCI-20041C-3	**High Performance Carrier Board**—Same as PCI-20041C-2 with the addition of full featured DMA supporting Analog I/O, Digital I/O and Counter/Frequency capability.

INSTRUMENT MODULES

PCI-20002M-1	**Analog Input Module**—16 single-ended or eight differential channels. 12-bit resolution and accuracy. Programmable gains of 1, 10, 100, 1K. Up to 32KHz sample rate.
PCI-20003M-2	**Analog Output Module**—High Speed. Two voltage outputs. Ranges: 0-10V, ±5V and ±10V FS. 12-bit resolution and accuracy. ±0.5LSB linearity. 3μs settling time.
PCI-20003M-4	**Analog Output Module**—High-Speed. Two voltage/current outputs. Ranges: 0-10V, ±5V, ±10V and 4-20mA FS. 12-bit resolution and accuracy.
PCI-20004M-1	**Digital I/O Module**—32 channels. TTL levels. Programmable I/O in groups of eight channels. Buffered outputs provide up to 24mA sink and 15mA source current. Input current is 200μA, max.
PCI-20005M-1	**Analog Input Expansion Module**—Adds 32 single-ended or 16 differential channels to the PCI-20002M-1. Inputs are protected 20V above power supplies.
PCI-20006M-1	**Analog Output Module**—High-Speed. One voltage output. 16-bit resolution and accuracy. Ranges: 0-10V, ±5V, and ±10V FS. ±.002% FS linearity. 10μs settling time.
PCI-20006M-2	**Analog Output Module**—High-Speed. Two Voltage outputs. 16-bit resolution and accuracy. Ranges: 0-10V, ±5V, and ±10V FS. ±.002% FS linearity. 10μs settling time.
PCI-20007M-1	**Counter/Timer/Pulse Generator Module**—Four Counter/Timer input channels, count and measure frequency. One programmable output pulse generator. ±.008% accuracy. 8MHz clock rate.
PCI-20017M-1	**Simultaneous Sample/Hold Module**—Four differential input channels. 20ns channel to channel scatter. Programmable gains: 1, 10, 100, 1000. 90db Common Mode rejection. Up to 30KHz bandwidth. 100μs settling time.
PCI-20019M-1	**High Speed Analog Input Module**—Eight channels. Single-ended. 12-bit resolution & accuracy. Up to 89kHz sample rate. Internal/external trigger. Automatic input channel advance. Compatible with PCI-20031M-1 Expander.
PCI-20020M-1	**Trigger/Alarm Module**—3.5μs response time. Dual channel. High, Low or Window comparisons. Built-in hysteresis. Reference programmable from −10V to +9.92V with 78mV resolution. Both analog and digital outputs available.
PCI-20021M-1	**Analog Output Module**—Eight voltage outputs. 12-bit resolution and accuracy. 0.5LSB linearity. Ranges: ±5 and ±10V FS. 500μs settling time. On-board data memory.
PCI-20023M-1	**High Speed Analog Input Module**—Eight channels, single-ended. 12-bit resolution and accuracy. Up to 180kHz sample rate. Internal/external trigger. Automatic input channel advance. Compatible with PCI-20031M-1 Expander.
PCI-20031M-1	**High Speed Analog Expansion Module**—Provides 32 high-speed channels with either the PCI-20002M-1 or PCI-20019M-1. On-board channel list memory. Inputs protected to 20V above power supplies.

TERMINATION PANELS

PCI-20010T-1	**Analog Signal Termination Panel**—16-channel, screw terminal connections. Can be used for inputs or outputs. Full passive signal conditioning capability. Provisions for external current loop connections.
PCI-20010T-2	**Analog Signal Termination Panel**—Optimized for thermocouple applications, but can be used for any analog input. Seven channels plus cold-junction compensation. Full passive signal conditioning capability. Provisions for external current loop connections.
PCI-20011T-1	**Digital Signal Termination Panel**—16-channel, screw-terminal connections. Provisions for LEDs, pullups, pulldowns, debounce filters, etc. Can be used for inputs or outputs.
PCI-20018T-1	**Isolated, Digital Termination Panel**—Eight-channel, screw terminal connections. Compatible with the PCI-1100 series and other industry standard opto isolators. LED indicators show activated channels. For inputs or outputs.

PCI-20042T-1 **Isolated, Analog Input Signal Conditioning Termination Panel**—Four channels (can be expanded with PCI-20043T-1). 750V channel-to-channel and input-to-output isolation. Differential inputs. Programmable gains: 1, 10, 100, 1000. Provisions for bridges and thermocouples. Bridge excitation.

PCI-20043T-1 **Isolated, Analog Input Expansion Termination Panel**—Adds four channels to PCI-20042T-1. Other features are the same. Units stack together and take only one physical space.

PCI-20044T-1 **Active, Analog Input, Signal Conditioning Termination Panel**—Four channels (can be expanded with PCI-20045T-1). Differential inputs. Programmable gains: 1, 10, 100, 1000. Provisions for bridges and thermocouples. Bridge excitation.

PCI-20045T-1 **Actiove, Analog Input Expansion Termination Panel**—Adds four channels to PCI-20044T-1. Other features are the same. Units stack together and take only one physical space.

PCI-20048T-1 **Isolated, Digital Termination Panel**—Same as PCI-20018T-1 except 16 channels. Compatible with PCI-20051A-1 enclosure and PCI-20052A-1 cover.

PCI-20057T-1 **High Density, Analog Signal Termination Panel**—48-channel, screw terminal connections. Passive signal conditioning capability. Thermocouple, cold-junction monitor. Can be used for inputs or outputs.

PCI-20058T-1 **High Density, Digital Signal Termination Panel**—48-channel, screw terminal connections. Provisions for pullups, pulldowns, debounce filters, etc. Can be used for inputs or outputs.

CABLES

PCI-20012A-1 **Analog Signal Cable**—Flat ribbon cable, six feet (2 meters) long. Fully shielded. Can be used with input or output devices.

PCI-20012A-2 **Analog Signal Cable**—Same as PCI-20012A-1 except 12 feet (4 meters) long.

PCI-20013A-1 **Digital Signal Cable**—Flat ribbon cable, six feet (2 meters) long. Ground-plane type shield. Can be used for input and output. For all digital termination panels except PCI-20058T-1.

PCI-20013A-2 **Digital Signal Cable**—Same as PCI-20013A-1 except 12 feet (4 meters) long. For all digital panels except PCI-20058T-1.

PCI-20032A-1 **Analog Output Cable**—Special purpose, three connector cable intended to connect up to three, one or two channel analog output modules to one termination panel. Not shielded. Six feet (2 meters) long.

PCI-20061A-1 **Digital Signal Cable**—Flat ribbon cable, six feet (2 meters) long. Ground-plane type shield. For use with PCI-20058T-1 termination panel. Can be used for input and output.

PCI-20062A-1 **Inter-Carrier Cable**—Flat ribbon cable, five inches (13cm) long. Ground-plane type shield. For use in inter-connecting (extending) the I^3 Bus between adjacent PCI-20041C carriers.

PCI-20062A-2 **Inter-Carrier Cable**—Flat ribbon cable, ten inches (25cm) long. Ground-plane type shield. For use in inter-connecting (extending) the I^3 Bus between non-adjacent PCI-20041C carriers.

OPTO-ISOLATORS

PCI-1101 **Digital Opto-Isolation Module**—10-32V AC/DC input. Isolation rating is 4000V. Compatible with PCI-20018T-1 and PCI-20048T-1 panels. One channel per module.

PCI-1102 **Digital Opto-Isolation Module**—90-140VAC/DC input. Isolation rating is 4000V. Compatible with PCI-20018T-1 and PCI-20048T-1 panels. One channel per module.

PCI-1103 **Digital Opto-Isolation Module**—5-60VDC output at 3A. Isolation rating is 4000V. Compatible with PCI-20018T-1 and PCI-20048T-1 panels. One channel per module.

PCI-1104 **Digital Opto-Isolation Module**—12-140VAC output, at 3A. Isolation rating is 4000V. Compatible with PCI-20018T-1 and PCI-20048T-1 panels. One channel per module.

PCI-1105 **Digital Opto-Isolation Module**—180-280V AC/DC input. Isolation rating is 4000V. Compatible with PCI-20018T-1 and PCI-20048T-1 panels. One channel per module.

PCI-1106 **Digital Opto-Isolation Module**—24-280VAC output at 3A. Isolation rating is 4000V. Compatible with PCI-20018T-1 and PCI-20048T-1 panels. One channel per module.

ENCLOSURES

PCI-20029A-1 **Quad Termination Panel Enclosure**—Rack or table top use. Holds up to four standard-size termination panels.

PCI-20051A-1 **Termination Panel Enclosure**—Rack mount enclosure for PCI-20048T-1 panel.

PCI-20052A-1 **Enclosure, Cover**—Cover for PCI-20051A-1 enclosure.

MISCELLANEOUS HARDWARE

PCI-20028A-3 **Strain-Relief Bracket**—Supports ribbon cables at rear of PC. In most applications, one is recommended for each Carrier used.

PCI-20033A-1 **Module Extender**—Facilitates calibration of PCI-20002M-1.

PCI-20038A-1 **DC Power Supply**—±15V at 0.8A. ±0.05% regulation. Short-circuit protected. 120VAC input (nominal).

PCI-20038A-3 **DC Power Supply**—±15V at 0.8A. ±0.05% regulation. Short—circuit protected. 240VAC input (nominal).

INTELLIGENT DATA LOGGER

PCI-20056K-1 **PCI ControLOGraph**—Hardware/software system for data-logger type applications. Completely menu driven. No computer skills required. Complete with all necessary hardware and software (except PC).

PCI-20054S-1 **Demonstration Software Diskette**—Presents a general overview of the ControLOGraph intelligent data logger system and its capabilities (see PCI-20056K-1).

SOFTWARE

PCI-20034S-1 **Demonstration Software Diskette**—Presents a general overview of the PCI-20000 system and its capabilities.

PCI-20040S-1 **LABTECH NOTEBOOK**—A menu-driven data acquisition, control and analysis software package. Works with many PCI-20000 hardware items. Provides real-time display and graphics (see PCI-20064S-1 below).

PCI-20046S-1 **Software Drivers**—BASIC language support subroutine library. Provides an uncomplicated, high performance interface between PCI-20000 hardware and the programmer.

PCI-20046S-2 **Software Drivers**—C language support subroutine library. Provides an uncomplicated, high performance interface between PCI-20000 hardware and the programmer.

PCI-20046S-3 **Software Drivers**—Turbo-Pascal language support subroutine library. Provides an uncomplicated, high performance interface between PCI-20000 hardware and the programmer.

PCI-20046S-4 **Software Drivers**—ASYST language support subroutine library. Provides an uncomplicated, high performance interface between PCI-20000 hardware and the programmer.

PCI-20046S-6 **Software Drivers**—Combination Package. Includes the PCI-20046S-1, PCI-20046S-2 and PCI-20046S-3.

PCI-20047S-1 **Software Drivers**—High speed and DMA extension to the PCI-20046S language support subroutine library.

PCI-20054S-1 **Demonstration Software Diskette**—Presents a general overview of the ControLOGraph intelligent data logger system and its capabilities (see PCI-20056K-1).

PCI-20064S-1 **Demonstration Software Diskette**—Presents a general overview of LABTECH NOTEBOOK and its capabilities (see PCI-20071S-1 below).

PCI-20065S-1 **REAL TIME ACCESS**—An extension to Labtech Notebook that supports real-time data transfers to spreadsheets or other analysis programs.

PCI-20067S-1 **DADiSP**—A menu driven, graphical, data analysis package.

PCI-20068S-1 **SNAPSHOT STORAGE SCOPE**—A menu driven waveform capture system. Useful for transient analysis and general digital oscilloscope applications (see PCI-20069S-1 below).

PCI-20069S-1 **Demonstration Software Diskette**—Presents a general overview of SNAPSHOT STORAGE SCOPE and its capabilities.

PCI-20072S-1 **Demonstration Software Diskette**—Presents an overview of DADiSP and its capabilities.

PCI-20073S-1 **Relay Ladder Logic RD1000/PC**—Software for process control and monitoring.

TABLE 10.2A. SPEED SUMMARY TABLE

This table is Not a specification sheet. Execution speed is a function of many factors, some of which are beyond the scope of a simple table. The speeds indicated here are offered as guidelines to assist the user in estimating the appropriateness of the PCI-20000 in a given application. All data is expressed in **Readings/Second**, unless otherwise noted, and includes the time to read or write to the PC's RAM using PCI software drivers.

BASIC (INTERPRETED) LANGUAGE

PARAMETER	CONDITIONS	IBM PC	IBM AT
Analog Input	PCI-20046S-1	Note 1	Note 2
	Channel Configuration		
	PCI-20002M-1 (READ.CH)		
	G = 1	92	400
	G = 10	92	400
	G = 100	91	357
	G = 1K	84	277
	with PCI-20005M-1's (READ.CH)		
	G = 1	92	400
	G = 10	92	400
	G = 100	91	357
	G = 1K	84	277
	with PCI-20017M-1[3] (READ.SSH)	330	1080
	Read Thermocouple (READ.CH)	80	238
	Read RTD (READ.CH)	77	212
	PCI-20019M-1 (READ.CH)	93	416
	with PCI-20031M-1 (READ.CH)	93	416
	with PCI-20017M-1[3] (READ.SSH)	348	1376
Analog Output	PCI-20046S-1 (WRITE.CH)	Note 1	Note 2
	PCI-20003M-2	90	415
	PCI-20003M-4	90	415
	PCI-20021M-1	90	415
	PCI-20006M-1 or -2	90	415
Digital I/O	PCI-20046S-1 (READ/WRITE.CH)	Note 1	Note 2
	PCI-20004M-1 or On-Carrier (Bytes/Sec)	125	435
Counter/Timer	PCI-20046S-1	Note 1	Note 2
	PCI-20007M-1 Read Counter (READ.CH)	125	435
	Read and Reset (STAT.CTN)	125	435
	Read Group[4] (READ.CTS)	114	430

BASIC (COMPILED) LANGUAGE

PARAMETER	CONDITIONS	IBM PC	IBM AT
Analog Input	PCI-20046S-1	Note 1	Note 2
	Channel Configuration		
	PCI-20002M-1 (READ.CH)		
	G = 1	769	2326
	G = 10	769	2326
	G = 100	714	1667
	G = 1K	455	667
	with PCI-20005M-1's (READ.CH)		
	G = 1	769	2326
	G = 10	769	2326
	G = 100	714	1667
	G = 1K	455	667
	with PCI-20017M-1[3] (READ.SSH)	1333	2500
	Read Thermocouple (READ.CH)	204	400
	Read RTD (READ.CH)	333	625
	PCI-20019M-1 (READ.CH)	833	3448
	with PCI-20031M-1 (READ.CH)	883	3448
	with PCI-20017M-1[3] (READ.SSH)	3636	14,285
Analog Output	PCI-20046S-1 (WRITE.CH)	Note 1	Note 2
	PCI-20003M-2	909	3571
	PCI-20003M-4	909	3571
	PCI-20021M-1	909	3571
	PCI-20006M-1 or PCI-20006M-2	833	3448
Digital I/O	PCI-20046S-1 (READ/WRITE.CH)	Note 1	Note 2
	PCI-20004M-1 or On-Carrier (Bytes/Sec)	770	3226
Counter/Timer	PCI-20046S-1	Note 1	Note 2
	PCI-20007M-1 Read Counter (READ.CH)	770	3226
	Read and Reset (STAT.CTN)	770	3226
	Read Group[4] (READ.CTS)	435	833

NOTES: (1) All data referring to the "IBM PC" is based upon a standard 4.77MHz machine.

(2) All data referring to the "IBM AT" is based upon an 8MHz machine.

(3) Data for the PCI-20017M-1 takes into account that each READ.SSH command reads four channels.

(4) Read Group data takes into account that each READ.CTS command reads three channels.

TABLE 10.2B. SPEED SUMMARY TABLE

This table is Not a specification sheet. Execution speed is a function of many factors, some of which are beyond the scope of a simple table. The speeds indicated here are offered as guidelines to assist the user in estimating the appropriateness of the PCI-20000 in a given application. All data is expressed in **Readings/Second,** unless otherwise noted, and includes the time to read or write to the PC's RAM using PCI software drivers.

C LANGUAGE

PARAMETER	CONDITIONS	IBM PC	IBM AT
Analog Input	PCI-20046S-1	Note 1	Note 2
	Channel Configuration		
	PCI-20002M-1 (READ.CH)		
	G = 1	833	3448
	G = 10	833	3448
	G = 100	770	2326
	G = 1K	476	667
	with PCI-20005M-1's (READ.CH)		
	G = 1	833	3448
	G = 10	833	3448
	G = 100	770	2326
	G = 1K	476	667
	with PCI-20017M-1[3] (READ.SSH)	1380	2500
	Read Thermocouple (READ.CH)	208	400
	Read RTD (READ.CH)	345	625
	PCI-20019M-1 (READ.CH)	909	3571
	with PCI-20031M-1 (READ.CH)	909	3571
	with PCI-20017M-1[3] (READ.SSH)	4000	15,385
Analog Output	PCI-20046S-1 (WRITE.CH)	Note 1	Note 2
	PCI-20003M-2	1000	3846
	PCI-20003M-4	1000	3846
	PCI-20021M-1	1000	3846
	PCI-20006M-1 or -2	833	3571
Digital I/O	PCI-20046S-1 (READ/WRITE.CH)	Note 1	Note 2
	PCI-20004M-1 or On-Carrier (Bytes/Sec)	833	3448
Counter/Timer	PCI-20046S-1	Note 1	Note 2
	PCI-20007M-1 Read Counter (READ.CH)	833	3448
	Read and Reset (STAT.CTN)	833	3448
	Read Group[4] (READ.CTS)	1363	2000

TURBO PASCAL LANGUAGE

PARAMETER	CONDITIONS	IBM PC	IBM AT
Analog Input	PCI-20046S-1	Note 1	Note 2
	Channel Configuration		
	PCI-20002M-1 (READ.CH)		
	G = 1	714	2273
	G = 10	714	2273
	G = 100	667	1639
	G = 1K	435	654
	with PCI-20005M-1's (READ.CH)		
	G = 1	714	2273
	G = 10	714	2273
	G = 100	667	1639
	G = 1K	435	654
	with PCI-20017M-1[3] (READ.SSH)	1290	2353
	Read Thermocouple (READ.CH)	200	385
	Read RTD (READ.CH)	322	588
	PCI-20019M-1 (READ.CH)	770	2439
	with PCI-20031M-1 (READ.CH)	770	2439
	with PCI-20017M-1[3] (READ.SSH)	3333	10,526
Analog Output	PCI-20046S-1 (WRITE.CH)	Note 1	Note 2
	PCI-20003M-2	833	2632
	PCI-20003M-4	833	2632
	PCI-20021M-1	833	2632
	PCI-20006M-1 or -2	770	2439
Digital I/O	PCI-20046S-1 (READ/WRITE.CH)	Note 1	Note 2
	PCI-20004M-1 or On-Carrier (Bytes/Sec)	714	2273
Counter/Timer	PCI-20046S-1	Note 1	Note 2
	PCI-20007M-1 Read Counter (READ.CH)	714	2273
	Read and Reset (STAT.CTN)	714	2273
	Read Group[4] (READ.CTS)	1250	1900

NOTES: (1) All data referring to the "IBM PC" is based upon a standard 4.77MHz machine.

(2) All data referring to the "IBM AT" is based upon an 8MHz machine.

(3) Data for the PCI-20017M-1 takes into account that each READ.SSH command reads four channels.

(4) Read Group data takes into account that each READ.CTS command reads three channels.

TABLE 10.2C. SPEED SUMMARY TABLE

This table is Not a specification sheet. Execution speed is a function of many factors, some of which are beyond the scope of a simple table. The speeds indicated here are offered as guidelines to assist the user in estimating the appropriateness of the PCI-20000 in a given application. All data is expressed in **Readings/Second**, unless otherwise noted, and includes the time to read or write to the PC's RAM.

ASYST LANGUAGE

PARAMETER	CONDITIONS		IBM PC	IBM AT
Analog Input	PCI-20046S-1		Note 1	Note 2
	Channel Configuration			
	PCI-20002M-1	(READ.CH)		
	G = 1		270	833
	G = 10		270	833
	G = 100		263	770
	G = 1K		222	455
	with PCI-20005M-1's	(READ.CH)		
	G = 1		270	833
	G = 10		270	833
	G = 100		263	770
	G = 1K		222	455
	with PCI-20017M-1[3]	(READ.SSH)	741	1481
	Read Thermocouple	(READ.CH)	137	333
	Read RTD	(READ.CH)	185	370
	PCI-20019M-1	(READ.CH)	278	909
	with PCI-20031M-1	(READ.CH)	278	909
	with PCI-20017M-1[3]	(READ.SSH)	1143	3636
Analog Output	PCI-20046S-1	(WRITE.CH)	Note 1	Note 2
	PCI-20003M-2		286	909
	PCI-20003M-4		286	909
	PCI-20021M-1		286	909
	PCI-20006M-1 or -2		278	909
Digital I/O	PCI-20046S-1	(READ/WRITE.CH)	Note 1	Note 2
	PCI-20004M-1 or On-Carrier	(Bytes/Sec)	270	833
Counter/Timer	PCI-20046S-1		Note 1	Note 2
	PCI-20007M-1 Read Counter	(READ.CH)	270	833
	Read and Reset	(STAT.CTN)	270	833
	Read Group[4]	(READ.CTS)	638	1250

ASSEMBLY LANGUAGE

PARAMETER	CONDITIONS	IBM PC	IBM AT
Analog Input	Channel Configuration	Note 1	Note 2
	PCI-20002M-1 Single Channel Read		
	G = 1	16,130	26,316
	G = 10	16,130	26,316
	G = 100	16,130	26,316
	G = 1K	16,130	26,316
	Multi Channel with PCI-20005M-1's		
	G = 1	7000	7500
	G = 10	7000	7500
	G = 100	6000	6000
	G = 1K	1000	1000
	with PCI-20017M-1	6000	7300
	PCI-20019M-1	62,000	89,000
	with PCI-20031M-1	62,000	89,000
	with PCI-20017M-1	17,000	33,000
	PCI-20023M-1	62,500	120,000
Analog Output		Note 1	Note 2
	PCI-20003M-2	27,000	80,000
	PCI-20003M-4 (Current Output)	27,000	40,000
	PCI-20021M-1	2000	2000
	PCI-20006M-1 or PCI-20006M-2	27,000	80,000
Digital I/O		Note 1	Note 2
	PCI-20004M-1 or On-Carrier (Bytes/Sec)	40,000	80,000
Counter/Timer		Note 1	Note 2
	PCI-20007M-1 Read Counter	30,000	90,000
	Read and Reset	24,390	62,100

NOTES: (1) All data referring to the "IBM PC" is based upon a standard 4.77MHz machine.

(2) All data referring to the "IBM AT" is based upon an 8MHz machine.

(3) Data for the PCI-20017M-1 takes into account that each READ.SSH command reads four channels.

(4) Read Group data takes into account that each READ.CTS command reads three channels.

TABLE 10.2D. SPEED SUMMARY TABLE
LANGUAGE INDEPENDENT—HIGH SPEED COMMANDS

This table is Not a specification sheet. Execution speed is a function of many factors, some of which are beyond the scope of a simple table. The speeds indicated here are offered as guidelines to assist the user in estimating the appropriateness of the PCI-20000 in a given application. All data is expressed in **Readings/Second**, unless otherwise noted, and includes the time to read or write to the PC's RAM using PCI software drivers.

PARAMETER	CONDITIONS	IBM PC	IBM AT
Analog Input	PCI-20047S-1, Block Mode[1] (HS.RUN)	Note 3	Note 4
	PCI-20002M-1 (mode 1)		
	G = 1	8000	9000
	G = 10	8000	9000
	G = 100	5500	5500
	G = 1K	970	970
	with PCI-20005M-1 or PCI-20031M-1 (mode 1)		
	G = 1	6900	7400
	G = 10	6900	7400
	G = 100	5500	5500
	G = 1K	970	970
	with PCI-20017M-1[5] (mode 2)	6000	7300
	PCI-20019M-1 (mode 1)	12,000	28,000
	with PCI-20031M-1 (mode 1)	12,000	26,000
	with PCI-20005M-1 (mode 1)	12,000	26,000
	with multiple MUX (mode 1)	10,000	25,000
	with two PCI-20017M-1's 8Ch[5] (mode 2)	17,000	33,000
	PCI-20019M-1 (mode 3)	15,000	32,000
	with PCI-20031M-1 (mode 3)	15,000	32,000
	with PCI-20005M-1 (mode 3)	15,000	32,000
	PCI-20019M-1[7] (mode 4)	62,500	89,000
	with PCI-20031M-1[2] and PCI-20020M-1 (mode 4)	62,500	89,000
	PCI-20023M-1 with PCI-20020M-1 (mode 4)	62,500	120,000
	PCI-20047S-1 (DMA.RUN)	Note 3	Note 4
	PCI-20002M-1 (1 Channel only)	32,000	32,000
	with PCI-20031M-1 (multi Chs)[2]		
	G = 1	12,350	12,350
	G = 10	12,350	12,350
	G = 100	7,600	7,600
	G = 1K	970	970
	PCI-20019M-1 with or without PCI-20031M's		
	Transfer Conditions:[6]		
	1K frames at 2 bytes/frame	89,000	60,000
	1K frames at 8 bytes/frame	89,000	82,000
	Continuous at 2 bytes/frame	89,000	46,000
	Continuous at 8 bytes/frame	89,000	62,000
	PCI-20023M-1 with or without PCI-20031M-1		
	Transfer Conditions:[6]		
	1K frames at 2 bytes/frame	180,000	60,000
	1K frames at 8 bytes/frame	180,000	82,000
	Continuous at 2 bytes/frame	180,000	46,000
	Continuous at 8 bytes/frame	180,000	62,000
Analog Output	PCI-20047S-1 (DMA.RUN)	Note 3	Note 4
	PCI-20003M-2		
	Transfer Conditions:[6]		
	1K frames at 3 bytes/frame	120,000	40,000
	Continuous at 3 bytes/frame	83,000	30,000
	PCI-20003M-4 (Current Output)	40,000	40,000
	PCI-20021M-1	2,000	2,000
	PCI-20006M-1 and PCI-20006M-2	120,000	53,000
	Transfer Conditions:[6]		
	1K frames at 3 bytes/frame	120,000	40,000
	Continuous at 3 bytes/frame	83,000	30,000
Digital I/O	PCI-20047S-1 (DMA.RUN)	Note 3	Note 4
	Transfer Conditions:[6]		
	1K frames at 2 bytes/frame	360Kbyte	120Kbyte
	1K frames at 8 bytes/frame	360Kbyte	164Kbyte
	Continuous at 2 bytes/frame	250Kbyte	91Kbyte
	Continuous at 8 bytes/frame	340Kbyte	124Kbyte
Counter/Timer	PCI-20047S-1 (DMA.RUN)	Note 3	Note 4
	PCI-20007M-1 Read Counter		
	Transfer Conditions:[6]		
	1K frames at 2 bytes/frame	180,000	60,000
	Continuous at 2 bytes/frame	125,000	45,000

NOTES: (1) Block Mode. This corresponds to reading one or more channels, one or more channels, one or more times.

(2) When the PCI-20031M-1 Expander Module is used with either HS.RUN, Mode 4 (i.e., with PCI-20019M) or DMA, analog input (i.e., with PCI-20002M or PCI-20019M), the total available channels are determined by the PCI-20031Ms. That is, the channels on the PCI-20002M or PCI-20019M are not available for use.

(3) All data referring to the "IBM PC" is based upon a standard 4.77MHz machine.

(4) Two different types of "IBM AT" computers are referred to in this data sheet. Block mode, analog input data is based upon an 8MHz machine, while all DMA data is based upon a 6MHz machine.

(5) Data for the PCI-20017M-1 takes into account that each READ.SSH command reads four channels.

(6) A "frame" is defined as the list containing the channels to be read (or written to). "1K frames" refers to reading the list 1024 times. "Bytes/frame" refers to the amount of data in each frame (reading one PCI-20019M-1, requires two bytes).

(7) In the Auto Channel Mode a trigger is required to insure proper channel identification.

THE PCI-20000 CONFIGURATION GUIDE

INTRODUCTION

The PCI-20000 is an intelligent instrumentation front end, designed to turn the personal computer (PC) into a powerful system for data acquisition, test, measurement, and control. The system resides on a printed circuit board that plugs into a convenient expansion slot within the host computer. The internal PCI-20000 architecture is designed to interface with most microcomputers. Compatibility with a specific PC is achieved with bus translation circuitry located on the plug-in board. Direct connection is made to the PC's internal bus, allowing high speed data acquisition and control.

Figure 10.1 is a block diagram showing the system's basic configuration. The unique design of the PCI-2000 allows the input/output configuration to be optimized for a particular application. The key concepts embodied within the PCI-20000 system include its modular construction, its proprietary "Intelligent Instrumentation Interface Bus" (I^3 Bus—patent pending) and its memory-mapped address structure. Figure 10.2 depicts the I^3 Bus structure.

Mechanically, the PCI-20000 system consists of several types of printed circuit boards: "Carriers," "Instrument Modules" and "Termination Panels." A carrier is the system's "main board." It is the carrier that plugs into one of the PC's expansion slots. The instrument modules connect, "piggyback" style, to the carrier. In general, each different module supports a unique I/O function. Many of the carriers have built-in I/O capabilities. That is, they have a useful role without any modules installed. The carriers and modules usually reside inside the PC or in an adjacent expansion enclosure. Thermination panels, on the other hand, are installed in external cabinets or on available mounting surfaces.

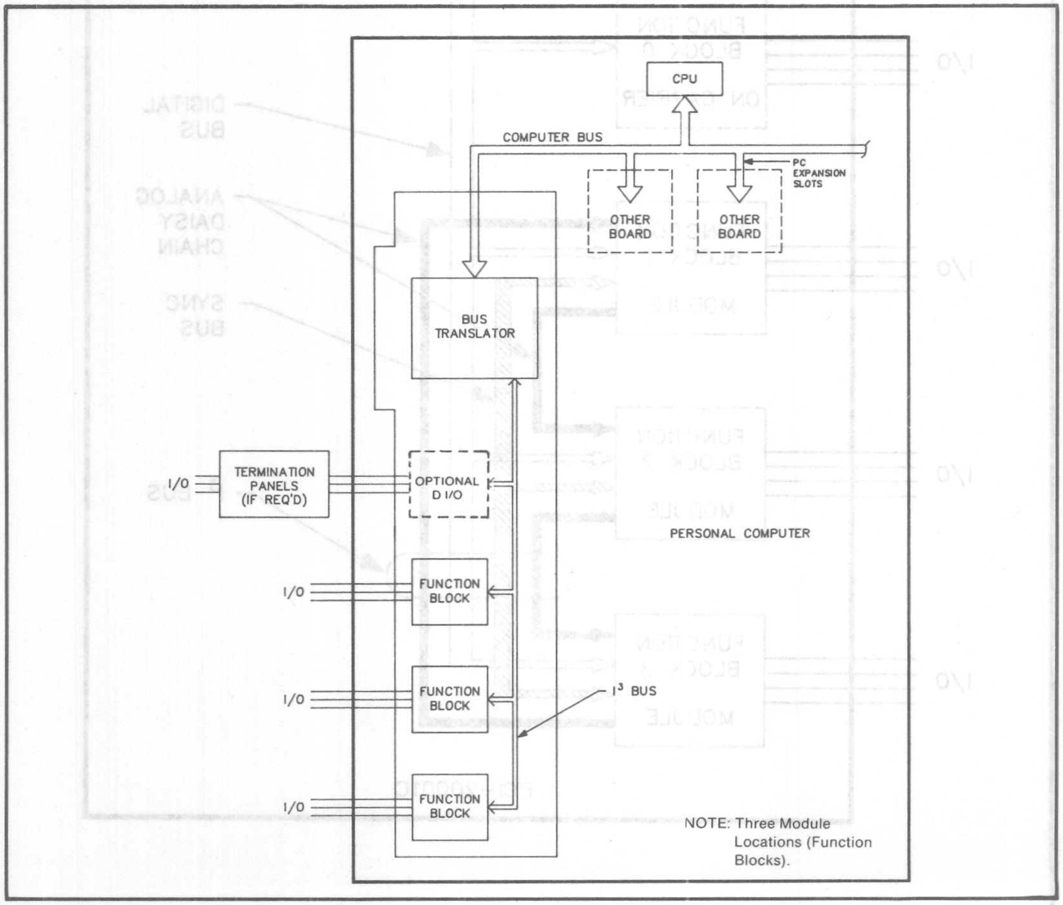


FIGURE 10.1. Personal Computer-Based Data Acquisition and Control System—Block Diagram.

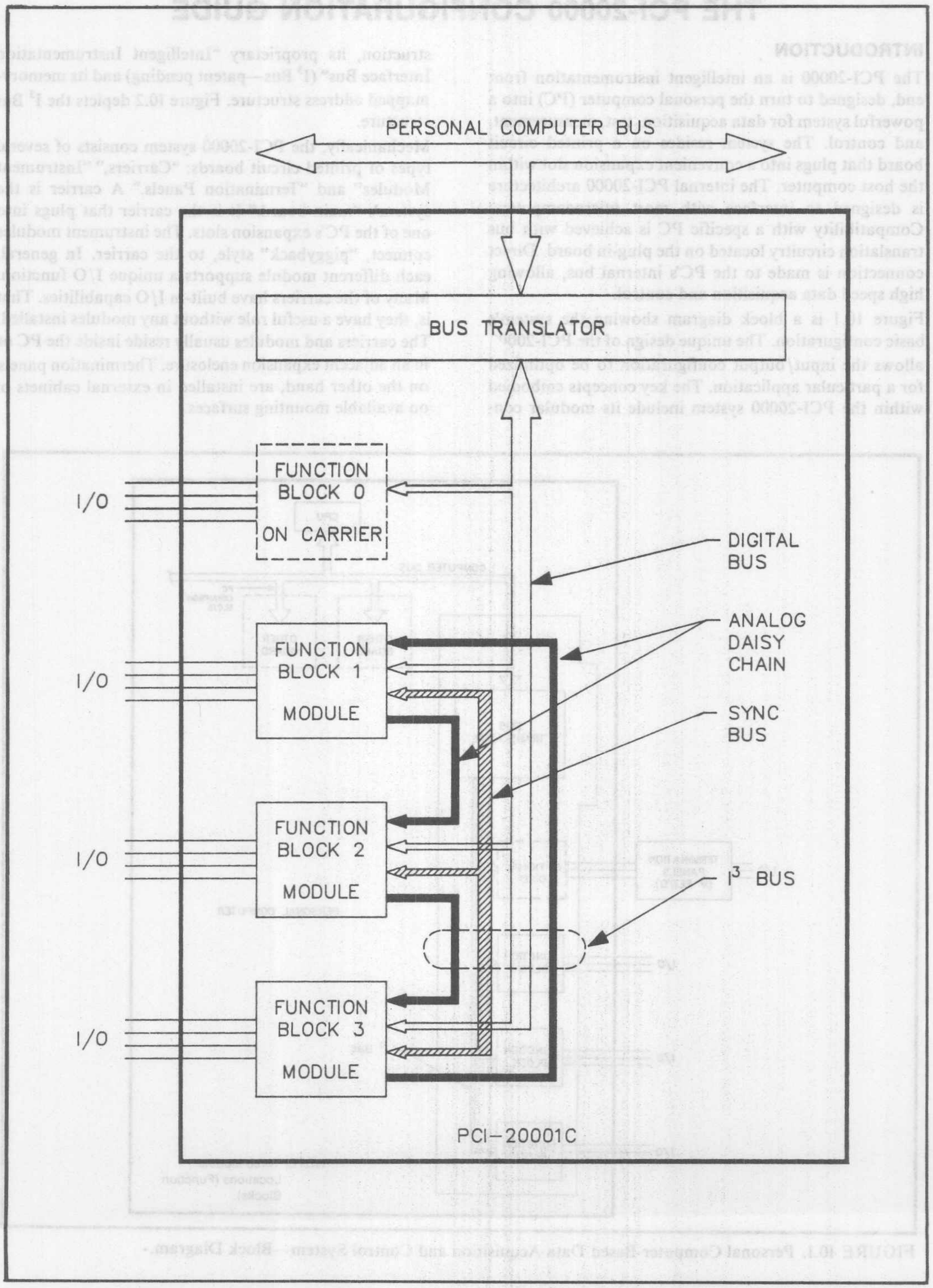


FIGURE 10.2. PCI-20000 I^3 Bus Structure.

The family of carriers now contains four types. These four types are designed for the IBM series of PCs as well as the COMPAQ, AT&T, Zenith, Siemens and other IBM-compatible personal computers. Each carrier provides space for up to three modules. Also included is the Intelligent Instrumentation Interface (I^3 Bus), which facilitates inter-module communications. Most carriers provide 32 digital I/O points as well as other useful features.

A wide variety of instrument modules for analog and digital applications can be obtained. The versatility and ease of expansion of a modular system based upon the I^3 bus mean that the family of modules will continue to grow. In addition, modules can be treated as building-block components in a wide range of applications not associated with standard PCs.

Signal termination panels within the system complement it by providing convenient screw-terminal connections between the internal electronics and the external field signals. Appropriate cables can be obtained to link the termination panels to the modules. Rack and table-top enclosures for the termination panels are also available.

Many software support packages already interface with the PCI-20000. Some of the packages contain a family of routines that can be called by high-level commands. Versions to work with BASIC, C, Turbo Pascal, ASYST and Assembly languages are now available. Complete applications packages for instrumentation, test, laboratory, and control tasks are also available. These different software products give the user easy access to the extensive features of the system.

Please refer to the PCI-20000 Specification Summary and to Figure 10.3 (PCI-20000 System Configuration Chart) for an overview of the many components within the PCI-20000 family. More details also appear in the indivudual Product Data Sheets shown later in this section.

CONFIGURING A PCI-20000 SYSTEM

This material is intended as an aid in configuring a system for a given application. As might be expected, it is often more difficult to define than to implement a project. However, all that is needed is a knowledge of the desired tasks and this handbook. Here are some starting suggestions:

TABLE 10.3. CONFIGURATION OUTLINE

A) Define the application's input/output requirements:
The number of analog inputs?
Single-ended or differential?
The number of digital outputs?
Analysis and control?
Display?
Speed? .... etc.

B) Define available resources:
Dollar budget?
Time available?
Skills and expertise of system integrators?
Skills of operators? .... etc.

C) Select appropriate Software drivers or software applications packages.

D) Select appropriate PCI-20000 Instrument Modules **in conjunction with** appropriate Active Signal Conditioner Termination Panel requirements.

E) Select appropriate PCI-20000 Carriers.

F) Select appropriate Termination Panels, Cables and Enclosures.

G) Select appropriate accessories (i.e., brackets, power supplies, etc.

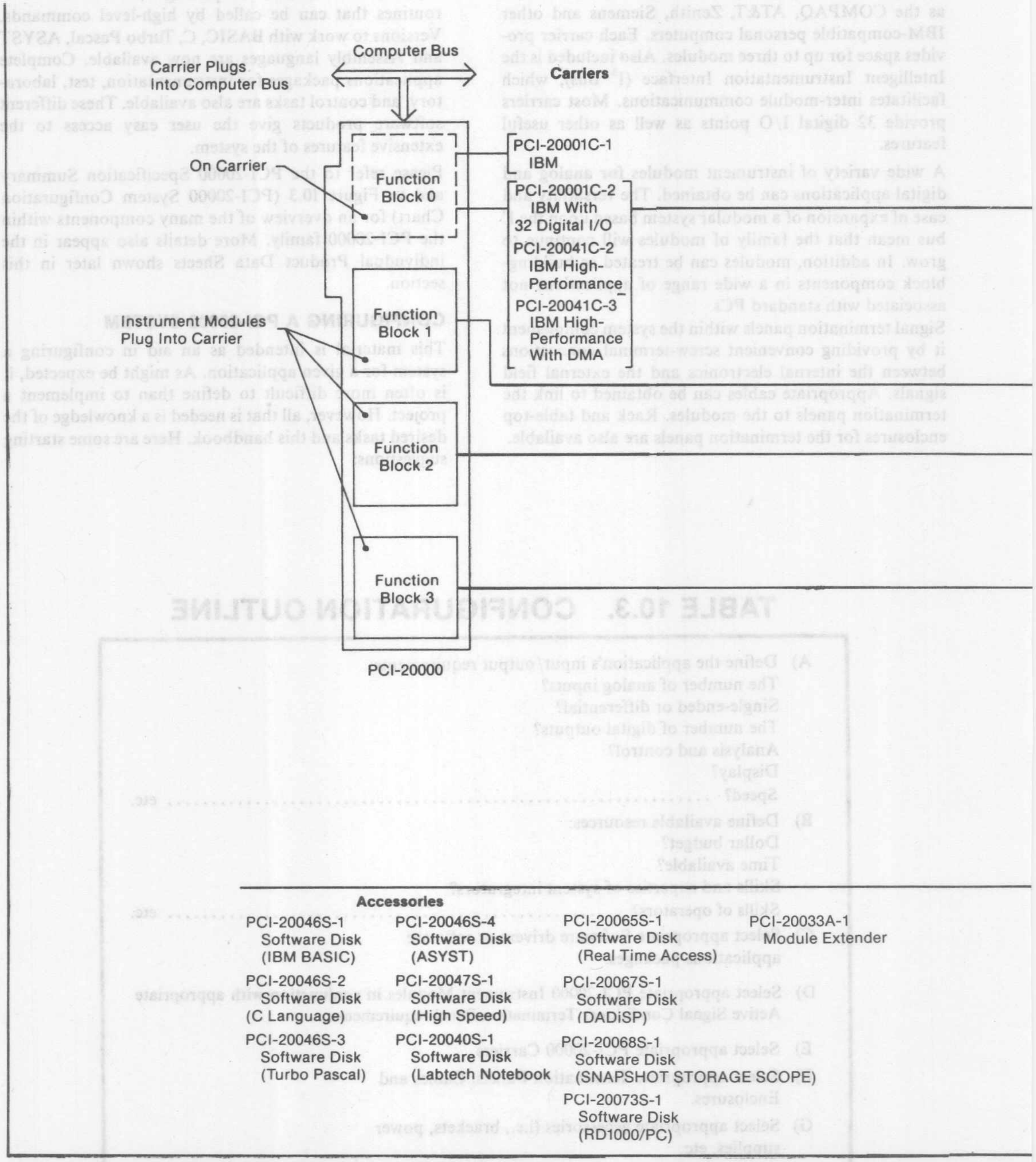
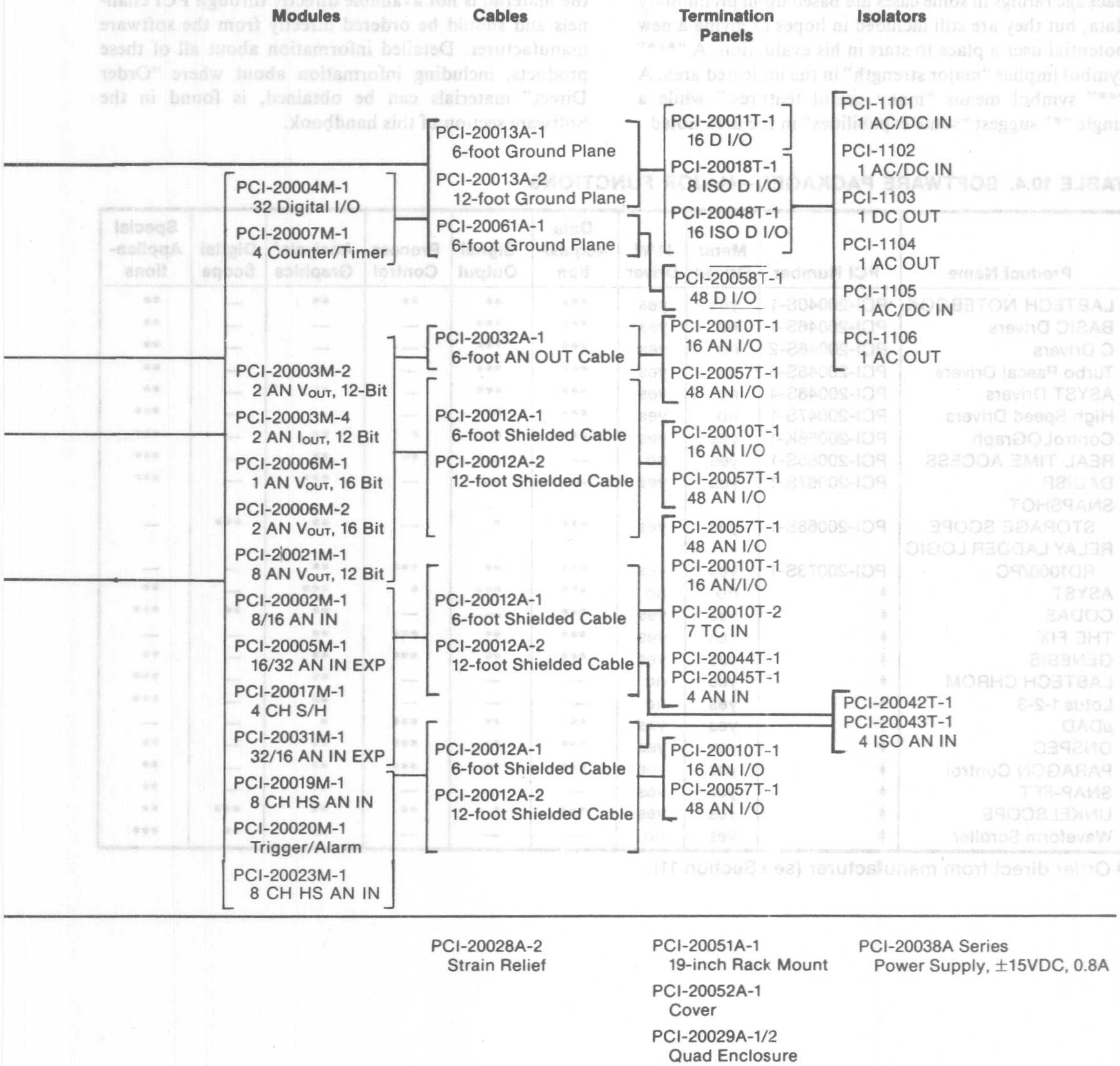


FIGURE 10.3. PCI-20000 System Component Configuration Chart.


Modules
Cables
Termination Panels
Isolators
PCI-20004M-1 32 Digital I/O
PCI-20007M-1 4 Counter/Timer
PCI-20003M-2 2 AN VOUT, 12-Bit
PCI-20003M-4 2 AN IOUT, 12 Bit
PCI-20006M-1 1 AN VOUT, 16 Bit
PCI-20006M-2 2 AN VOUT, 16 Bit
PCI-20021M-1 8 AN VOUT, 12 Bit
PCI-20002M-1 8/16 AN IN
PCI-20005M-1 16/32 AN IN EXP
PCI-20017M-1 4 CH S/H
PCI-20031M-1 32/16 AN IN EXP
PCI-20019M-1 8 CH HS AN IN
PCI-20020M-1 Trigger/Alarm
PCI-20023M-1 8 CH HS AN IN
PCI-20013A-1 6-foot Ground Plane
PCI-20013A-2 12-foot Ground Plane
PCI-20061A-1 6-foot Ground Plane
PCI-20032A-1 6-foot AN OUT Cable
PCI-20012A-1 6-foot Shielded Cable
PCI-20012A-2 12-foot Shielded Cable
PCI-20012A-1 6-foot Shielded Cable
PCI-20012A-2 12-foot Shielded Cable
PCI-20012A-1 6-foot Shielded Cable
PCI-20012A-2 12-foot Shielded Cable
PCI-20011T-1 16 D I/O
PCI-20018T-1 8 ISO D I/O
PCI-20048T-1 16 ISO D I/O
PCI-20058T-1 48 D I/O
PCI-20010T-1 16 AN I/O
PCI-20057T-1 48 AN I/O
PCI-20010T-1 16 AN I/O
PCI-20057T-1 48 AN I/O
PCI-20057T-1 48 AN I/O
PCI-20010T-1 16 AN/I/O
PCI-20010T-2 7 TC IN
PCI-20044T-1
PCI-20045T-1 4 AN IN
PCI-20010T-1 16 AN I/O
PCI-20057T-1 48 AN I/O
PCI-1101 1 AC/DC IN
PCI-1102 1 AC/DC IN
PCI-1103 1 DC OUT
PCI-1104 1 AC OUT
PCI-1105 1 AC/DC IN
PCI-1106 1 AC OUT
PCI-20042T-1
PCI-20043T-1 4 ISO AN IN
PCI-20028A-2 Strain Relief
PCI-20051A-1 19-inch Rack Mount
PCI-20052A-1 Cover
PCI-20029A-1/2 Quad Enclosure
PCI-20038A Series Power Supply, ±15VDC, 0.8A

SOFTWARE CONFIGURATION TABLE

A summary of the major software produts obtainable to operate with the PCI-20000 is shown below. Software package ratings in some cases are based upon preliminary data, but they are still included in hopes of giving a new potential user a place to start in his evaluation. A "***" symbol implies "major strength" in the indicated area. A "**" symbol means "many useful features," while a single "*" suggest "some capabilities" in the area noted.

Note that some of these products are listed with PCI numbers while others are not. "Order Direct" means that the material is not available directly through PCI channels and should be ordered directly from the software manufacturer. Detailed information about all of these products, including information about where "Order Direct" materials can be obtained, is found in the Software section of this handbook.

TABLE 10.4. SOFTWARE PACKAGES—MAJOR FUNCTIONS

Product Name	PCI Number	Menu Driven	H/W Driver	Data Acquisition	Signal Output	Process Control	Analysis/ Graphics	Digital Scope	Special Applications
LABTECH NOTEBOOK	PCI-20040S-1	yes	yes	***	**	**	**	—	**
BASIC Drivers	PCI-20046S-1	no	yes	***	***	—	—	—	**
C Drivers	PCI-20046S-2	no	yes	***	***	—	—	—	**
Turbo Pascal Drivers	PCI-20046S-3	no	yes	***	***	—	—	—	**
ASYST Drivers	PCI-20046S-4	no	yes	***	***	—	**	—	**
High Speed Drivers	PCI-20047S-1	no	yes	***	***	—	—	—	***
ControLOGraph	PCI-20056K-1	yes	yes	***	*	*	**	—	***
REAL TIME ACCESS	PCI-20065S-1	yes	no	—	—	**	**	—	***
DADiSP	PCI-20067S-1	yes	yes	—	—	—	***	—	***
SNAPSHOT STORAGE SCOPE	PCI-20068S-1	yes	yes	***	*	—	**	***	—
RELAY LADDER LOGIC RD1000/PC	PCI-20073S-1	yes	yes	***	**	***	**	—	—
ASYST	‡	no	no	***	***	*	***	—	**
CODAS	‡	yes	yes	***	—	—	**	**	***
THE FIX	‡	yes	yes	***	**	***	**	—	—
GENESIS	‡	yes	yes	***	**	***	**	—	**
LABTECH CHROM	‡	yes	no	—	—	—	**	—	***
Lotus 1-2-3	‡	yes	no	—	—	—	**	—	***
μDAD	‡	yes	yes	***	**	***	*	—	—
ONSPEC	‡	yes	yes	***	**	***	**	—	**
PARAGON Control	‡	yes	yes	***	**	***	**	—	**
SNAP-FFT	‡	yes	yes	—	—	—	**	—	**
UNKELSCOPE	‡	yes	yes	***	*	**	**	***	**
Waveform Scroller	‡	yes	no	—	—	—	—	**	***

‡ Order direct from manufacturer (see Section 11).

HARDWARE CONFIGURATION TABLES

The next table offers a speed summary of the various Analog Input/Expander combinations. "Total Channels" refers to the selected configurations (Single-Ended/Differential). "Hardware Speed" denotes the maximum capabilities of the hardware modules (Readings/Sec). "PCI S/W Speed" reflects the system's attainable performance using the PCI-20046S/47S software drivers (Readings/Sec). An IBM PC (running at 4.77MHz) and an IBM PC/AT (running at 6MHz) are assumed.

Tables for other components of the PCI-20000 System follow. These tables are Not specification sheets. Execution speed is a function of many factors, some of which are beyond the scope of a simple table. The speeds indicated here are offered as guidelines to assist the user in estimating the appropriateness of the PCI-20000 in a given application.

TABLE 10.5. ANALOG INPUT AND MULTIPLEXER—SPEED SUMMARY TABLE

Configuration	Total Channels	Hardware Speed	PCI S/W Speed (PC)	PCI S/W Speed (AT)
NORMAL MODE (All Carriers)				
PCI-20002M-1, alone (Gain = 1)	16/8	13kHz	8kHz	9kHz
	1/1	32kHz	16kHz	26kHz
PCI-20002M-1 and PCI-20005M-1 or PCI-20031M-1	48/24	13kHz	7kHz	8kHz
PCI-20019M-1, alone	8	89kHz	62kHz	89kHz
PCI-20019M-1 and PCI-20005M-1	40	32kHz	15kHz	32kHz
PCI-20019M-1 and PCI-20031M-1	32	89kHz	62kHz	89kHz
PCI-20023M-1, alone or with PCI-20031M-1	8	180kHz	62kHz	120kHz
DMA MODE (PCI-20041C-3 only)				
PCI-20002M-1, alone (Gain = 1)	1/1	32kHz	32kHz	32kHz
PCI-20002M-1 and PCI-20031M-1	32/16	14kHz	12kHz	12kHz
PCI-20019M-1, alone(1)	8	89kHz	89kHz	82kHz
PCI-20019M-1 and PCI-20019M-1	16	178kHz	178kHz	82kHz
PCI-20019M-1 and PCI-20031M-1	32	89kHz	89kHz	82kHz
PCI-20023M-1, alone or with PCI-20031M-1	8	180kHz	180kHz	82kHz

NOTE: (1) The DMA transfer speed in an IBM PC/AT can be highly dependent upon the amount of data transferred. Please refer to the Speed Summary Tables in the previous section for more information.

TABLE 10.6. ANALOG OUTPUT MODULES—SUMMARY TABLE

Module Type	Number of Channels	V_{OUT}	I_{OUT} 4-20mA	Max Speed(1)	Resolution	Linearity
PCI-20003M-2	2	Yes	No	80K	12 bits	.5LSB
PCI-20003M-4	2	Yes	Yes	40K(2)	12 bits	1LSB
PCI-20006M-1	1	Yes	No	80K	16 bits	.002%FS
PCI-20006M-2	2	Yes	No	80K	16 bits	.002%FS
PCI-20021M-1	8	Yes	No	2K	12 bits	.5LSB

NOTES: (1) Output points/sec in an IBM PC/AT computer. (2) The PCI-20003M-4 is jumper programmable so that either channel can be used for voltage or current output. The specs shown are for the current mode. For the voltage mode see the PCI-20003M-2.

TABLE 10.7. SPECIAL PURPOSE MODULES—SUMMARY TABLE

Module Type	Function	Number of Channels	Special Features
PCI-20004M-1	Digital I/O	32	TTL levels, programmable in bytes (8 bits), buffered outputs
PCI-20007M-1	Counter/Timer	5	4 independent counters (8MHz), 1 programmable clock generator
PCI-20017M-1	Simultaneous Sample/Hold	4	Differential Inputs, BW = 30kHz, G = 1-1K, 20ns channel-to-channel scatter
PCI-20020M-1	Trigger/Alarm	2	Window mode (1 channel) hi-low mode, 3.5μs response time

TABLE 10.8. CARRIER—SUMMARY TABLE

Carrier Type	I^3 Bus On Carrier	Inter-Carrier Extension of I^3 Bus	Digital I/O	DMA	External Sync/ Interrupt	Pacer Clock
PCI-20001C-1	Yes	No	No	No	No	No
PCI-20001C-2	Yes	No	Yes	No	No	No
PCI-20041C-2	Yes	Yes	Yes	No	Yes	Yes
PCI-20041C-3	Yes	Yes	Yes	Yes	Yes	Yes

TABLE 10.9. TERMINATION PANELS AND SIGNAL CONDITIONERS—SUMMARY TABLE

Panel Type	Function	Number Channels	T.C.	Bridge	Conditioning Active	Conditioning Passive	Mating Enclosures
PCI-20010T-1	Analog I/O	16/8	No	No	No	Yes	PCI-20029A-1
PCI-20057T-1	Analog I/O	48/24	Yes	No	No	Yes	PCI-20029A-1
PCI-20010T-2	Analog/Thermocouple	0/7	Yes	No	No	Yes	PCI-20029A-2
PCI-20042T-1	Isolated Active Conditioner	0/4	Yes	Yes	Yes	Yes	PCI-20029A-1
PCI-20043T-1	Isolated Active Expander	0/4	Yes	Yes	Yes	Yes	PCI-20029A-1
PCI-20044T-1	Active Signal Conditioner	0/4	Yes	Yes	Yes	Yes	PCI-20029A-1
PCI-20045T-1	Active Signal Expander	0/4	Yes	Yes	Yes	Yes	PCI-20029A-1
PCI-20011T-1	Digital I/O	16	No	No	No	Yes	PCI-20029A-1
PCI-20058T-1	Digital I/O	48	No	No	No	Yes	PCI-20029A-1
PCI-20018T-1	OPTO-Digital I/O	8	No	No	Yes(1)	No	PCI-20029A-1
PCI-20048T-1	OPTO-Digital I/O	16	No	No	Yes(1)	No	PCI-20051A-1

NOTE: (1) Individual opto-isolator modules are required for each channel. See a description of the PCI-1100 Series below.

TABLE 10.10. DIGITAL OPTO-ISOLATION MODULES—SUMMARY TABLE

Module Type	Function	Input Range	Output Range	No. Chn.	Isolation Rating
PCI-1101	AC/DC Input	10–32V	TTL	1	4000V
PCI-1102	AC/DC Input	90–140V	TTL	1	4000V
PCI-1105	AC/DC Input	180–280V	TTL	1	4000V
PCI-1103	DC Output	TTL	5–60V/3A	1	4000V
PCI-1104	AC Output	TTL	12–140V/3A	1	4000V
PCI-1106	AC Output	TTL	24–280V	1	4000V

TABLE 10.11. CABLES—SUMMARY TABLE

Cable Type	Analog/ Digital	Number of Channels	Length	Shield	Mating Modules PCI-200XXX	Mating Terminations PCI-200XXX
PCI-20012A-1	Analog	16/8	6ft (2m)	Yes	2M, 3M, 5M, 6M, 17M, 19M, 20M, 21M, 31M	10T, 42T, 44T, 57T
PCI-20012A-2	Analog	16/8	12ft (4m)	Yes	2M, 3M, 5M, 6M, 17M, 19M, 20M, 21M, 31M	10T, 42T, 44T, 57T
PCI-20032A-1	Analog	6	6ft (2m)	No	3M, 6M	10T, 57T
PCI-20013A-1	Digital	16	6ft (2m)	Yes	1C, 41C, 4M, 7M	11T, 18T, 48T
PCI-20013A-2	Digital	16	12ft (4m)	Yes	1C, 41C, 4M, 7M	11T, 18T, 48T
PCI-20061A-1	Digital	16	6 ft (2m)	Yes	1C, 41C, 4M, 7M	58T
PCI-20062A-1	Special	—	5in (13cm)	Yes	Inter-Carrier	
PCI-20062A-2	Special		10in (25cm)	Yes	Inter-Carrier	

SYSTEM EXPANSION GUIDELINES—

The minimum useful PCI-20000 system configuration consists of a single carrier with on-board digital I/O. This system requires only one expansion slot within the host PC. When modules are installed onto a carrier, the added thickness of the modules may result in mechanical intrusion into the adjacent expasion slot area provided for another expasion board. Whether this interference actually does occur depends upon the computer used, the number of modules installed, the location of the modules and the length of the board in the adjacent slot. If for example, a short board (Async, IEEE-488, etc.) is installed next to a PCI-20000 carrier with one module inserted in the "front" position, no interference is encountered. In this case, the PCI-20000 is still a single slot device. This could also be true with two modules installed. When all three module locations are occupied on a carrier, the system normally requires the space of two expansion slots. Please note, however, that the functional capability of this type of installation greatly exceeds that of any other combination of products that occupy only one or two slots. For example, a single carrier with three modules can provide 80 analog inputs or 128 digital I/O channels.

POWER REQUIREMENTS

Most modern personal computers provide several expansion slots for optional user selected boards. A partial listing of IBM compatible machines and estimates of their available expansion slots can be found in Section 2 of this handbook. The PC's power supply is designed to provide a "reasonable" amount of current to the expansion slots. The amount of power available is determined by subtracting the base systems power requirements from the total power supply rating. Please consult your computer's documentation for this information. Each carrier and module in the PCI-20000 system has its power requirements specified on its data sheet. It is a simple matter to sum the individual terms (current times voltage) to determine the total load. Some modules require $\pm15V$ power. These levels are not attainable directly from the computer. Each carrier has a DC/DC converter, on-board, to generate these voltages from the computer's raw +5V line. When determining total power requirements, be sure to select the "Equivalent" +5V current which appears at the bottom of each module's specification sheet. This number includes the conversion efficiency of the DC/DC converter. Because of the thousands of possible PCI-20000 configurations and the different computer power supply ratings, it is not practical to generate a compatibility table. In general, however, adequate power is available for all boards that will physically fit in a given PC.

SAMPLE CONFIGURATION EXAMPLE

In order to demonstrate Table 10.3 (Configuration Outline), a hypothetical example will be defined and an appropriate system configuration will be found.

Given:

- A manufacturing plant produces "widgets." In addition to process control, both the quantity and quality of the product must be monitored on all three production lines.
- Each line consists of a conveyor belt, a four zone furnace, two raw material flow controllers, two raw material weighing scales, four control valves, six product position indicators and 12 status indicator lights.
- The control algorithm requires five PID loops and four On-Off loops. On-screen graphics display of product output and major system parameters is required.

SOLUTION (Please refer to Table 10.3)

A) Define the I/O Requirements

Evaluation of the above facts along with additional information obtained after further investigation yields these I/O requirements:

1 — Frequency Input (conveyor speed)
4 — Thermocouple Inputs (furnace temperature)
5 — Analog Outputs (furnace and speed control)
2 — 4 to 20mA Analog Inputs (flow rate)
2 — 200Ω strain gage (scales)
6 — Digital Inputs (position switches)
16 — Digital Outputs (valves and indicators)

- The frequency input is in the range of 5kHz to 10kHz.
- The T.C. inputs must be isolated for 440VAC.
- All digital inputs are dry switch contacts.
- The 12 "status light" digital outputs must switch 240VAC at 1A.
- The existing motor controller accepts 0-10V inputs.
- The existing furnace controllers accept 0-10V inputs.
- All inputs and outputs must be updated once each second.

B) Define Resources

- System will be designed and implemented by "Henry."
- Henry is a process engineer, but he has NO programming experience.
- Mechanical and electrical installers are available.
- The system operators are unskilled.
- The system must be operational in six weeks.

C) Select Software

Because Henry has no programming experience, one of the available "Applications Packages" will be selected.

Refer to Table 10.4 SOFTWARE PACKAGES. Several products including LABTECH NOTEBOOK satisfy the above requirements, Other possible choices include LOOPWORKS, THE FIX, GENESIS, ONSPEC and PARAGON Control. Note that these products are menu driven (no programming is required) and they support data acquisition, process control, data analysis and graphics display. For this example LABTECH will be selected.

D) Select Modules and Active Signal Conditioner Termination Panels

Refer to the appropriate HARDWARE CONFIGURATION SUMMARY TABLEs.

1 — Frequency Input. Select a PCI-20007M-1 module.

4 — Thermocouple (TC) Inputs. Because isolation is required, a PCI-20042T-1 will be selected. This signal conditioner termination panel supports all four TCs. Select a PCI-20002M-1 (A/D) module to accept the PCI-20042T-1 output.

5 — Analog Outputs. Several choices exist. Select the PCI-20021M-1 (the most space- and cost-efficient).

2 — Strain Gage Inputs. The single-element strain gages employed require both bridge completion and excitation. Isolation is not required. Select a PCI-20045T-1 to mate with the PCI-20042T-1 and thus provide up to four additional input channels with the desired characteristics. Note that another A/D module is not required.

2 — 4 to 20mA Inputs. While normally not required, the most efficient way to accommodate these inputs is to use the spare channels on the existing PCI-20045T-1. No additional hardware is required.

6 — Digital Inputs. A carrier can be selected with 32 digital I/O points on-board. This option will be selected.

18 — Digital Outputs. Each eight-bit group of digital I/O can be configured as either inputs or outputs. Therefore, the remainder of the carrier's digital I/O ports will be utilized as outputs. No additional hardware is required.

E) Select Carriers

Three instrument modules (total) have been selected in addition to four digital I/O ports (eight bits each). The PCI-20001C-2 provides the required digital I/O and accommodates the three modules.

F) Select Termination Panels, Cables and Enclosures

In addition to the two signal conditioner panels selected above, terminations for the analog outputs, frequency input and digital I/O are also required. Remember to refer to Figure 10.3 (Configuration Chart) for information about the compatibility of the various components.

5 — Analog outputs. Select the PCI-20010T-1 termination panel. A PCI-20012A-1 cable connects the panel to the PCI-20021M-1.

1 — Frequency Input. Select the PCI-20058T-1 termination panel. This panel has three connectors and accommodates a total of 48 channels (two bytes per connector). Many of the extra terminals will be used for the required digital I/O. A PCI-20061A-1 cable connects the panel to the PCI-20007M-1.

6 — Digital Inputs. The second bank of the above PCI-20058T-1 panel will be used for this purpose. Provisions exist on the panel to install "pull-up" resistors to

"wet" the input switch contacts. A PCI-20061A-1 cable connects the panel to one of the two digital I/O connectors on the carrier.

4 — Digital Outputs. These are for the valve controls. The second bank of the above PCI-20058T-1 panel is shared for this purpose. Remember that each digital I/O connector on the carrier accommodates two bytes (16 channels). Each byte can be programmed as either inputs or outputs. Therefore, the same cable that is used for the digital inputs also supports this digital output function.

12 Digital Outputs. These are for the process indicators. Recall that these outputs must switch 240VAC at 1A. To accomplish this, digital opto isolators will be used. The PCI-1106 module meets the requirements. A separate opto module is used for each channel. The modules plug into the PCI-20048T-1 termination panel. A PCI-20013A-1 cable connects the panel to the second digital I/O connector on the carrier.

Select Enclosures — A total of five termination panels have been selected. From the standpoint of mounting space, the PCI-20042T-1 and the PCI-20045T-1 are one panel (they stack). These panels as well as the PCI-20010T-1 and the PCI-20058T-1 fit inside one PCI-20029A-1 enclosure (one spare space remains). A cover is included with this enclosure. The PCI-20048T-1 is a different style panel and is accommodated by the PCI-20051A-1 Enclosure. The mating cover is the PCI-20052A-1.

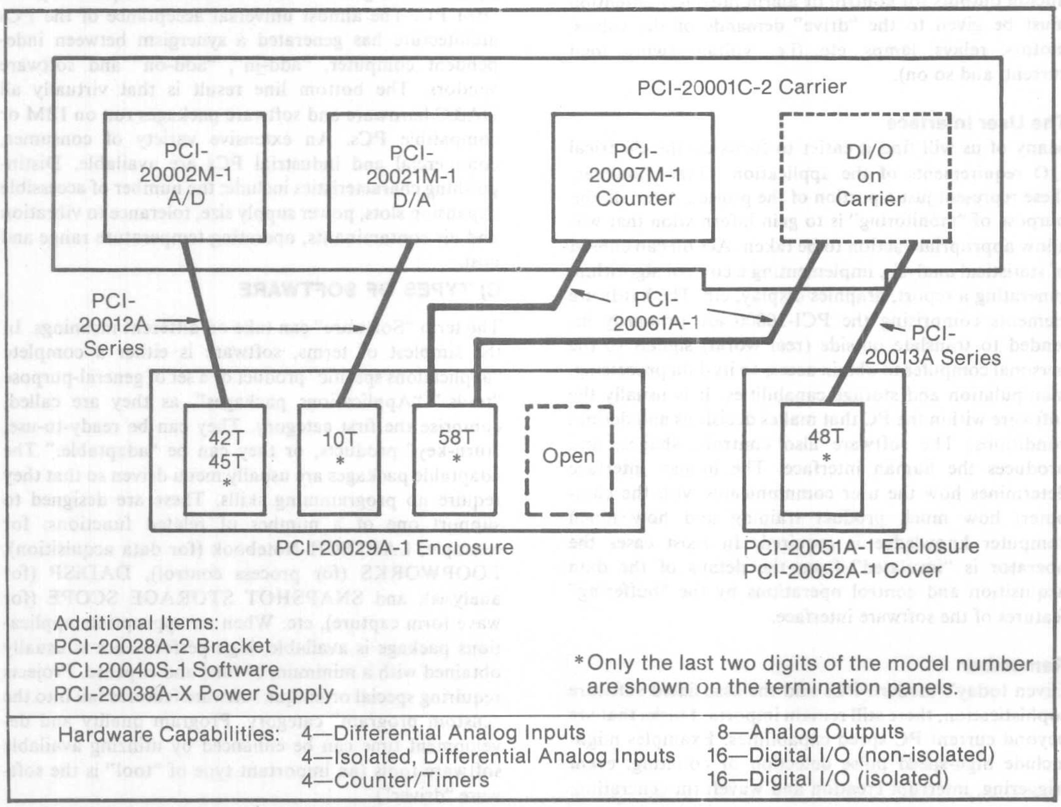
G) Select Accessories

Referring again to Figure 10.3 (Configuration Chart), two accessories are noted.

1 — A Power Supply is required to provide a ±15V to the active signal conditioner termination panels. The PCI-20038A series is selected (choose the appropriate dash number depending upon the required AC line voltage). The power supply can mount either inside or on the rear of the PCI-20029A-1 Enclosure.

1 — An additional strain relief bracket is recommended to support the cables exiting the rear of the computer. Each carrier comes with one bracket. To make assembly as easy as possible, it is suggested that only the digital I/O cables be routed through this clamp. The remaining cables should be clamped with a PCI-20028A-2 accessory bracket. This bracket is positioned in the adjacent expansion slot's mounting location.

SAMPLE CONFIGURATION



ADDITIONAL INFORMATION—DETAILED ANALYSIS OF THE CONFIGURATION PROCESS

This material is provided for additional assistance to first time users of data acquisition and control systems. The notes below are keyed to Table 10.3, the Configuration Outline that was presented earlier in this section.

A) DEFINE THE APPLICATION

List the project requirements in terms of the electrical signals involved. Remember that temperature, pressure, displacement and speed, etc. will be converted to voltages or currents by sensors and transducers. While these devices are an important part of the "total" system, only their signals are significant in the selection and definition of the data acquisition and control components.

The distinguishing features of **every** input and output signal (current or voltage, amplitude, frequency components, dynamic range, resolution and accuracy requirements, etc.) must be clearly known. How many channels of each different type are required? Are the analog inputs to be treated as single-ended or differential? (As a general rule, voltage levels below 1 volt should use differential connections for high accuracy.) When producing outputs for control or alarm purposes, attention must be given to the "drive" demands of the valves, motors, relays, lamps, etc. (i.e., voltage swing, load current, and so on).

The User Interface

Many of us will find it easier to focus on the electrical I/O requirements of the application. Often, however, these represent just a portion of the project. After all, the purpose of "monitoring" is to gain information that will allow appropriate action to be taken. Action can consist of statistical analysis, implementing a control algorithm, generating a report, graphics display, etc. The hardware elements comprising the PCI-20000 are primarily intended to translate outside (real world) signals to the personal computer to obtain access to its data processing, manipulation and storage capabilities. It is usually the software within the PC that makes decisions and defines conditions. The software also controls, shapes, and produces the human interface. The human interface determines how the user communicates with the computer; how much product training and how much computer knowledge is required. In most cases the operator is "insulated" from the details of the data acquisition and control operations by the "buffering" features of the software interface.

Remember

Given today's modern PCs and the associated software sophistication, there still remain important tasks that are beyond current PC speed capabilities. Examples might include high-speed pulse detection or counting, event triggering, interrupt creation and waveform generation. Fortunately, there are ways to enhance the overall system performance. In the PCI-20000 approach, "hardware solutions" are often available to solve or circumvent personal computer and software limitations.

B) ASSESSING RESOURCES

Who are you? What is your background? Are you familiar with computers, electronics, writing software, installing cable, interfacing transducers, etc.? If not, will these skills be required on this particular job? Who can be recruited to join your team? Look to your vendors not only for the quality of their products but also for their commitment to customer support, both before and after the sale.

Hardware Versus Software

A computerized data acquisition and/or control system consists of several major parts. These include the computer hardware, the input/output hardware and the software. It is important to remember that all hardware must be directed or controlled by software. In fact, there is an interdependence that renders either essentially useless without the other. So then, which is selected first—hardware or software? For many this question can invoke a spirited debate. However, when specifying a DA&C system, the choice is made much simpler by the significant degree of standardization inspired by the IBM PC. The almost universal acceptance of the PCs architecture has generated a synergism between independent computer, "add-in", "add-on" and software vendors. The bottom line result is that virtually all DA&C hardware and software packages run on IBM or compatible PCs. An extensive variety of consumer, commercial and industrial PCs are available. Distinguishing characteristics include: the number of accessible expansion slots, power supply size, tolerance to vibration and air contaminants, operating temperature range and cost.

C) TYPES OF SOFTWARE

The term "Software" can take on different meanings. In the simplest of terms, software is either a complete "applications specific" product or a set of general-purpose "tools." "Applications packages", as they are called, comprise the first category. They can be ready-to-use, "turn-key" products, or they can be "adaptable." The adaptable packages are usually menu driven so that they require no programming skills. These are designed to support one of a number of related functions: for example; LABTECH Notebook (for data acquisition), LOOPWORKS (for process control), DADiSP (for analysis), and SNAPSHOT STORAGE SCOPE (for wave form capture), etc. When an appropriate applications package is available, high performance is usually obtained with a minimum of time and expense. Projects requiring special or unique software features fall into the "custom program" category. Program quality and development time can be enhanced by utilizing available software tools (an important type of "tool" is the software "driver").

Custom programs can take many forms. They can be written in low-level languages such as machine or assembly language. While offering the highest performance, low-level programming usually requires the largest investment in software skills and time. High-level languages, such as BASIC, C, and TURBO PASCAL, provide a friendlier environment for most people. The use of easy to learn mnemonics and other recognizable phrases, to represent complex operations, explains their popularity.

As mentioned above, "software drivers" assist in the development of custom programs. Drivers usually consist of a family of programs or subroutines. The drivers bridge the gap between the project programmer and the DA&C hardware. While drivers are actually separate programs, their features are attainable upon request of the main program. They are written and optimized, often in assembly language, by people who have expertise with the particular hardware. This relieves the requirement for the project programmer to become intimately familiar with the I/O system. Most popular high-level languages, as well as assembly language, can "call" these drivers to facilitate all of the PCI-20000 hardware functions.

D) SELECTING MODULES

Most of the time, it is an individual module (or carrier, in the case of digital I/O) that determines a channel's I/O features. There are, however, important characteristics that are dramatically altered by other devices: for example; converting a differential signal to single-ended, stripping away a high common-mode voltage (isolation), amplifying a low-level signal, converting a voltage to a current, and detecting a predetermined level crossing, etc. Signal conditioning operations such as these can result from preprocessing the input signal with an active signal conditioner termination panel or with another PCI module. What is the significance of this? Can an analog input voltage module (A/D module) with single-ended inputs and no amplifier, accurately process a thermocouple (producing a 10mV, diferential signal) attached to a motor winding at 440VAC? Absolutely, YES! The flexibility of the PCI-20000 accommodates this and many other difficult applications. The requirements of this example can be provided by an Active Signal Conditioner Termination Panel (PCI-20042T-1). Cold-junction monitoring is included along with differential inputs, signal gain, and complete input-to-output and channel-to-channel isolation. The output signal is ideally suited to any analog input module.

Single-Ended Versus Differential

The PCI-20002M-1 analog input module can be configured for either single-ended or differential use. All of the channels are set as a group, yielding either 16 or 8 channels, respectively. Naturally, a differential input can be used with a single-ended signal, but the opposite is not true. The need for just one differential channel, divides the original single-ended channel count by two. One alternative consists of using the PCI-20020M-1 (Simultaneous Sample/Hold Module) as a preamplifier to the PCI-20002M-1. The PCI-20020M-1 has a separate instrumentation amplifier for each of its four channels. These can convert differential inputs to single-ended, without altering the original 16 channels provided by the PCI-20002M-1. However, as described in subsection F below, there are exceptions to consider: for example; the PCI-20042T-1 through PCI-20045T-1 (Active Signal Conditioner Termination Panels) also use on-board amplifiers to convert from differential to single-ended signals. Despite this, the number of input channels is limited to a total of eight for each **input module connector**. Because of the many possible combinations of PCI components, please be sure to account for the proper number of accessible channels.

Finding the Right Combination of Specifications

Depending upon the complexity of the application, the selection of modules may be an iterative process. Experience suggests that "speed" is often the most difficult parameter to satisfy. Therefore, channels requiring the fastest response should be specified first. Basic distinguishing information for each module can be found in the SPECIFICATION SUMMARY (Table 10.1). However, because there are many ways to interconnect the various PCI-20000 components, "complete specifications" must take into account the particular configuration. In particular, please refer to the SPEED SUMMARY TABLES (Table 10.2) including the MULTIPLEXER SPEED TABLE (Table 10.5). Figure 10.3 graphically shows all of the components that comprise the PCI-20000 system. Also indicated are the associated options and accessories. Compatibility of each group of components is indicated by the interconnecting lines.

Multiplexer Selection

Analog input multiplexers or expanders can be used as a low cost way of increasing the number of system channels. Figure 10.1 suggests how the analog portion of the I^3 bus can connect the multiplexer outputs to an existing input module (PCI-20002M-1 or PCI-20019M-1).

Two different types of Analog Input Expanders are available in the PCI-20000 system: the PCI-20005M-1 and the PCI-20031M-1. The PCI-20005M-1 is a general purpose multiplexer, primarily intended to extend the channel count of the PCI-20002M-1 Analog Input Module, while the PCI-20031M-1 is high-speed mux tailored to the PCI-20019M-1 (however, it can be used with the PCI-20002M-1). It is characteristic of the PCI-20005M-1 that it is "software driven." This means that the controlling program must issue separate instructions to the PCI-20005M-1 each time a new channel is to be selected. In contrast, the PCI-20031M-1 includes an automatic, internally driven, channel scanner that sequences through the desired inputs without computer intervention. The major consequence of this hardware

difference is that systems incorporating the PCI-20031M-1 can operate much faster than they can using the PCI-20005M-1.

There are several combinations of Input and Expander modules that are of particular interest. The user can decide from a variety of choices how best to optimize the system for a given task. Some arrangements offer the most channels at the lowest cost, while others provide the highest possible speed. It should be noted that, for a given hardware set, the total number of accessible channels can be dependent upon the maximum speed desired.

Digital Input/Output

Digital I/O function blocks (whether on a PCI-20004M-1 module or on a carrier) can be software configured for either input or output use, in groups of eight points (byte size). Thus, a 32-point module can be used for eight inputs and 24 outputs or 16 inputs and 16 outputs or 32 outputs.

E) SELECTING CARRIERS

Once the number of modules is known, it is a simple matter to determine the required carriers. Remember that each carrier has provisions for up to three modules. The choice among carriers is a funciton of desired features. The four currently available carriers are designated as either "General Purpose" (PCI-20001C series) or "High Performance" (PCI-20041C series).

By referring to the CARRIER-SUMMARY TABLE (Table 10.8) the major distinctions can easily be noted. A carrier with digital I/O is often selected, even if no initial digital requirements are known. Experience has shown, that in the long term, most applications will benefit from (if not require) digital I/O. Two major advantages are derived by including this capability on the carrier: lower cost and higher funtional density. The alternative to on-board digital I/O is to add a digital I/O module. While this is always a viable option, a separate module does have a higher cost per channel and consumes an otherwise open plug-in position on the carrier.

Pacer Clock

A Pacer Clock is a crystal controlled, fully programmable, timing generator. In some applications the computer's clock can be used to trigger interrupts, and to pace data acquisition, etc. This is described in the APPLICATIONS section of this handbook. However, at high speeds or when high stability is necessary an independent time-base is often required. This is the main purpose of the on-board pacer clock (PCI-20041C series), or alternately, the "Rate Generator" portion of the PCI-20007M-1 Counter/Timer Module. Again, including this function on the carrier lowers cost and saves space.

Inter-Carrier Bus

In "larger" applications the inter-carrier bus feature (an extension of the I^3 Bus), on the High Performance Carriers, allows up to five carriers to be connected together. Thus, a number of multiplexer modules on more than one carrier can "feed" one analog input module (A/D). This significantly contributes to the reduction in cost per channel for larger systems.

External Sync

"High Performance Carriers" also have a separate connector to provide support for external interrupts, start conversions, start/stop DMA and other related sychronization signals. With respect to DMA, ONLY the PCI-20041C-3 has DMA capabilities at the time of this publication. The proprietary DMA technique has distinct advantages and unique capabilities. These include the ability to transfer either input or output data from analog, digital and counter sources simultaneously, at speeds up to 360K bytes per second. In certain applications, performance limitations and constraints are imposed by the design of the PC itself. Most notable of these PC restrictions is that DMA transfers can only proceed in one direction at a time (input or output). Also, despite the higher clock rate of the PC/AT compared to the PC, internal computer circuit differences actually result in a reduction of the DMA transfer rate. Please refer to the SPEED SUMMARY TABLES (Table 10.2) and the Product Data Sheets for complete information.

F) SELECTING TERMINATION PANELS

Field signals are readily mated to the PCI-20000 system through the use of optional termination panels and signal conditioners. All panels offer easy-to-use screw connections and provide passive signal conditioning capabilities. Some panels are called signal conditioners and offer active functions such as amplification, isolation and bridge excitation. Panels are also available in different sizes to suit a range of applications.

Why Use Termination Panels?

The function and utility of termination panels are often understated. Termination panels are used as a connection point to the outside world. Why not avoid this intermediary step and connect the field signals directly? The answer to this question lies in the need for signal conditioning. These panels are designed to support a wide range of user installed signal conditioning networks. Appropriate units are tailored for voltage division, filtering, surge suppression, current to voltage conversion, contact wetting, etc. Extensive information on this subject is found elsewhere in this handbook. (Please see the section on signal conditioning.)

Cables, General

Each termination panel also has a cable connector which interfaces the panel to the appropriate PCI carrier or module. A family of cables is obtainable to support this purpose. The cables can be divided into three categories: analog I/O, digital I/O and special purpose. Analog cables are fully shielded, ribbon cables fitted with 26 pin connectors. Digital cables are "ground-plane" type ribbon

cables fitted with 34 pin connectors. As a general rule, each cable accommodates 16 channels. In the case of differential analog inputs, two physical channels on the input multiplexer are used for each signal. Therefore, one cable will support eight differential analog inputs. When a module hosts fewer than 16 channels (i.e., counter/timer, analog output, trigger/alarm) this rule does not apply. Rather, a complete cable will be used for that module regardless of the number of conductors actually used. There are, however, exceptions. For example, a high speed analog output module can provide up to two channels each. In some systems it is common for two or three of these modules to be used together on one carrier. Several cables and termination panels could be used to connect this configuration. Another option would be to use the special "three-headed" cable (PCI-20032A-1) which permits up to three, one- or two-channel modules to be conected to a single termination panel. Most cables can be obtained in both 6- and 12-foot lengths (2m and 4m). For those that wish to fabricate their own cables, full "pin-out" information is provided in the applicable user manuals.

Active Signal Conditioner Analog Termination Panels

There are two basic types of active analog signal conditioner termination panels. They have similar capabilities with the exception that one provides complete channel-to-channel isolation. Both families have precision, differential input, instrumentation amplifier front ends. The instrumentation amplifiers are jumper programmable for gains of 1, 10, 100, and 1000. There is a separate amplifier for each channel. Provisions have been made for "bridge completion" and each channel has its own adjustable excitation current source. Passive networks are also accommodated.

The amplifiers, isolators and power supplies on an active signal conditioner panel take considerably more area than simple passive networks alone. High voltage isolation imposes additional spacing requirements. As a result, the channel density (channels per board) and the cable configuration of these panels have some unique features. Each type consists of a pair of panels. One panel can be thought of as a "master" (PCI-20042T-1 or PCI-20044T-1) and the other an "expander" (PCI-20043T-1 or PCI-20045T-1). Each has four input channels. When up to four channels are required, a single PCI-20042T/44T is used alone. A connector on the panel mates with a standard PCI-20012A cable which couples with an analog input module. Under these conditions, the remaining input channels associated with the unused wires in the cable are not available for use. However, four more channels can be added by using the PCI-20043T/45T expander panels. The master/expander panels are designed to "stack", one above the other, to conserve mounting space. A short ribbon cable, supplied with the expanders, joins the two panels together. The result is that all eight channels are merged together in a single PCI-20012A cable going to the remainder of the system. Additional channel expansion on a single cable is not possible.

While the inputs to an active signal conditioner panel are differential, the outputs are always single-ended. Therefore, be sure that the mating analog input module is configured for single-ended operation. In applications where very high voltage gain or maximum bandwidth is desired, it is permissible to utilize the gain blocks on both the termination panel and the input module.

All analog, active signal conditioner termination panels require a source of external ±15V power. Please refer to the Signal Conditioner data sheets, the "accessory" notes below, and the PCI-20038A data sheet.

Opto-Isolation

The Digital Opto-Isolation Termination Panels (PCI-20018T-1 and PCI-20048T-1), require a separate PCI-1100 series module to be installed for each channel. Models for both AD and DC, Input and Output are available. The six different units are summarized below.

Enclosures

Termination panels are usually installed outside of the computer in existing facilities or in PCI enclosures. Two basic types of enclosures are now provided. Both are designed to be placed in standard 19-inch racks or in a table-top configuration. The so called "quad" enclosures can, in general, accommodate any four of the many PCI termination panels. One exception to this rule is the 16-channel digital opto-isolation panel. This large panel is housed in a companion enclosure that holds one each.

G) Selecting Accessories

When several cables are required to exit the PC, it is often convenient to make use of unused, expansion board mounting locations. An accessory strain relief bracket is well-suited to secure the ribbon cables and to help protect internal components (PCI-20028A-2). In general, one accessory bracket is recommended per carrier.

The active, analog, signal conditioning termination panels require an external source of ±15V power. Power supplies are available to serve this purpose. Two models cover the world-wide AC line input requirements (PCI-20038A-1 and PCI-20038A-3). The .8A output is sufficient for most applications. Each can be mounted inside or on the rear of the PCI-20029A Termination Panel Enclosures.

PCI-20000 PERSONAL COMPUTER INTERFACE SYSTEM TYPICAL CONFIGURATIONS

The PCI-20000 is the most advanced, flexible and powerful scientific and industrial personal computer interface system on the market. Virtually hundreds of different configurations are possible, all of which plug directly inside an IBM PC, XT, AT or other hardware-compatible personal computer or expansion box. The backbone of the PCI-20000 system is the Carrier board which plugs into a computer expansion slot. A Carrier "carries" up to three Modules which perform specific interface functions such as digital input/output, analog data acquisition, analog output, and so forth. In addition, the Carrier may also provide specific interface functions such as digital input/output. Modules and Carriers are selected to match specific applications. Signal Termination Panels, Cables, and Software complete a PCI-20000 system.

A key point to remember is that the PCI-20000 is modular—you can select just the right components needed for your application. Most of the following examples, which include some popular configurations, use only one Carrier board. Multiple PCI-20000 Carriers can be used to create large systems. Feel free to modify these examples to create the configuration you need for your application. The examples include:

Configuration "A"—128 points of digital input/output.

Configuration "B"—80 single-ended/40 differential analog inputs, plus an optional 32 points of digital input/output.

Configuration "C"—35 channels of thermocouple input, plus an optional 32 points of digital input/output.

Configuration "D"—8 channels of fast analog data acquisition with programmable rate control and trigger level, plus an optional 32 points of digital input/output.

Configuration "E"—16 single-ended/8 differential analog inputs, 2 analog outputs, 4 counter/timer/pulse generator channels, 1 rate generator, and 32 digital inputs/outputs.

Configuration "F"—6 channels of high-resolution analog output, plus an optional 32 points of digital input/output.

Configuration "G"—4 channel simultaneous sample/hold analog data acquisition with programmable triggering, plus an optional 32 points of digital input/output.

Configuration "H"—32 channels of analog input utilizing DMA and programmable triggering.

Configuration "I"—512 points of digital I/O under DMA control.

Configuration "J"—4 channels of analog waveform generation under DMA control.

CONFIGURATION "A"—128 POINTS OF DIGITAL INPUT/OUTPUT

This configuration provides 128 buffered TTL-level digital I/O points on a single Carrier board under program control. Each point can sink 24mA or source 15mA. Since this example is designed using the modular PCI-20000 System, substitutions can easily be made. For example, if only 48 digital input/output points are needed, two PCI-20004M-1 Digital Input/Output Modules, three PCI-20061A-1 Digital Cables and one PCI-20058T-1 Digital Termination Panel are required.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "A"

1 PCI-20001C-2 Carrier with Digital Input/Output (32 points).

3 PCI-20004M-1 Digital Input/Output Modules (32 points each).

8 PCI-20061A-1 Digital Input/output Cables (1 cable for each 16 points).

3 PCI-20058T-1 Digital Termination Panels (1 panel for each 48 points).

RECOMMENDED CONFIGURATION "A" OPTIONS

1 PCI-20046S-1 Software Support Package for Microsoft/IBM BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan ASYST.

1 PCI-20029A-1 Quad Termination Panel Enclosures.

1 PCI-20028A-2 Strain Relief Bracket.

If your application requires isolation of digital signals, you could use eight PCI-20048T-1 Opto-isolation Termination Panels instead of the PCI-20058T-1 Digital Termination Panels. These panels accept the PCI-1100 Series digital opto-isolator modules. A PCI-20048T-1 Panel accepts up to 16 opto-isolator modules. A PCI-20018T-1 Opto-isolation Termination Panel could be substituted, which accepts up to eight isolators. (The PCI-20048T-1 and PCI-20018T-1 require PCI-20013A series cables.)

One PCI-20029A-1 Quad Termination Panel Enclosure can house up to four PCI-20000 Series analog or digital termination panels. Separate enclosures, PCI-20051A-1, are available for each PCI-20048T-1 Panel.

The PCI-20028A-2 Strain Relief Bracket is used to securely fasten the input/output cables at the rear of the personal computer.

CONFIGURATION "B"—80 SINGLE-ENDED/40 DIFFERENTIAL ANALOG INPUTS, PLUS AN OPTIONAL 32 POINTS OF DIGITAL INPUT/OUTPUT

This configuration provides a total of 80 single-ended or 40 differential analog inputs with full 12-bit resolution and accuracy. Throughput is 8000 samples/second with a gain of 1 or 10. In addition to analog inputs, an optional 32 points of buffered digital input/output are also available. Since this configuration is designed using the modular PCI-20000 System, substitutions can easily be made. For example, if only 48 analog input channels were needed, one PCI-20005M-1 Analog Input Expansion Module, five PCI-20012A-1 Analog Cables and only one PCI-20057T-1 Termination Panel are required.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "B"

1 PCI-20001C-1 Carrier.
1 PCI-20002M-1 Data Acquisition Module (16 single-ended or 8 differential channels, 12-bit).
2 PCI-20005M-1 Analog Input Expansion Modules (32SE/8DIF channels).
5 PCI-20012A-1 Analog Shielded Cables (one cable required for every 16/8 channels).
2 PCI-20057T-1 Analog Termination Panels

RECOMMENDED CONFIGURATION "B" OPTIONS

1 PCI-20046S-1 Software Support Package for Microsoft/IBM BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan ASYST.
1 PCI-20029A-1 Quad Termination Panel Enclosures.
1 PCI-20028A-2 Strain Relief Bracket.

A PCI-20001C-2 Carrier could be substituted for the PCI-20001C-1 Carrier if 32 points of buffered TTL-level digital input/output are desired. Digital cables and termination panels would also be required.

The PCI-20028A-2 Strain Relief Bracket is used to securely fasten the input/output cables at the rear of the personal computer.

CONFIGURATION "C"—35 CHANNELS OF THERMOCOUPLE INPUT, PLUS AN OPTIONAL 32 POINTS OF DIGITAL INPUT/OUTPUT

This configuration includes all the necessary PCI-20000 System elements for 35 channels of thermocouple inputs, plus an optional 32 points of buffered TTL-level digital input/ouput. Since this configuration uses the modular PCI-20000 System, substitutions can easily be made. For example, if only 21 thermocouple channels are required, one PCI-20005M-1 Analog Expansion Module, three PCI-20012A-1 Shielded Analog Cables and one PCI-20057T-1 Termination Panels are required.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "C"

1 PCI-20001C-1 Carrier.
1 PCI-20002M-1 Data Acquisition Module (seven thermocouple channels plus one cold-junction compensation channel).
2 PCI-20005M-1 Analog Input Expansion Modules (15 thermocouple channels plus two cold-junction compensation channels).
5 PCI-20012A-1 Analog Shielded Cables (one cable required for every seven thermocouple channels).
2 PCI-20057T-1 Termination Panels.
1 PCI-20029A-2 Thermocouple Termination Panel Enclosure (one enclosure for every four termination panels required).

RECOMMENDED CONFIGURATION "C" OPTIONS

1 PCI-20046S-1 Software Support Package for Microsoft/IBM BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan ASYST.
1 PCI-20028A-2 Strain Relief Bracket.

All PCI-20046S series software contains J, K, and T thermocouple linearization. A PCI-20001C-2 Carrier could be substituted for the PCI-20001C-1 Carrier if 32 points of buffered digital input/output are desired. Digital cables and termination panels would also be required.

The PCI-20028A-2 Strain Relief Bracket is used to securely fasten the input/output cables at the rear of the personal computer.

CONFIGURATION "D"—8 CHANNELS OF FAST ANALOG DATA ACQUISITION WITH PROGRAMMABLE RATE CONTROL AND TRIGGER LEVEL, PLUS AN OPTIONAL 32 POINTS OF DIGITAL INPUT/OUTPUT

This configuration is capable of analog data acquisition up to 89K samples/second with an IBM PC AT. The sampling rate is controlled by a software-programmable pacer clock on one of the modules. A trigger module allows a signal that is "greater than," "less than," "inside a window" or "outside a window" to start the data acquisition process. An optional 32 points of buffered TTL-level digital input/output are available.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "D"

1 PCI-20001C-1 Carrier.
1 PCI-20019M-1 High-Speed Data Acquisition Module (8 single-ended channels, fast A/D converter).
1 PCI-20020M-1 Trigger/Alarm Module (programmable triggering).
1 PCI-20007M-1 Rate Generator Module (controls sample rate).

2 PCI-20012A-1 Analog Shielded Cables.
1 PCI-20057T-1 Analog Termination Panel.

RECOMMENDED CONFIGURATION "D" OPTIONS

1 PCI-20046S-1 Software Support Package for Microsoft/IBM BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan ASYST.
1 PCI-20047S-1 High Performance Software Support Package.
1 PCI-20029A-1 Quad Termination Panel Enclosures.
1 PCI-20028A-2 Strain Relief Bracket.

A PCI-20001C-2 Carrier could be substituted for the PCI-20001C-1 Carrier if 32 points of buffered TTL-level digital input/output are desired. Digital cables and termination panels would also be required.

One PCI-20029A-1 Quad Termination Panel Enclosure can house up to four analog or digital termination panels.

The PCI-20028A-2 Strain Relief Bracket is used to securely fasten the input/output cables at the rear of the personal computer.

CONFIGURATION "E"—16 SINGLE-ENDED/8 DIFFERENTIAL ANALOG INPUTS, 2 ANALOG OUTPUTS, 4 COUNTER/TIMER/PULSE GENERATOR CHANNELS, 1 RATE GENERATOR, AND 32 DIGITAL INPUT/OUTPUTS

This configuration provides an excellent general purpose interface for a personal computer. Included are 16 single-ended or 8 differential analog inputs with 12-bit resolution and accuracy; two 12-bit analog voltage/current outputs; 32 points of buffered digital input/output; and 4 counter/timer/pulse generator channels, and one rate generator. This configuration could be modified to fit a particular application by eliminating certain elements or substituting others.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "E"

1 PCI-20001C-2 Carrier with Digital Input/Output (32 points).
1 PCI-20002M-1 Data Acquisition Module (16 single-ended/8 differential channels, 12-bit).
1 PCI-20003M-4 Dual Analog Output Module (12-bit, voltage or current outputs).
1 PCI-20007M-1 Counter/Timer/Pulser Module (one rate generator and four general-purpose channels).
2 PCI-20012A-1 Analog Shielded Cables (one required for the data acquisition module, one for the analog output module).
3 PCI-20013A-1 Digital Ground-plane Cables (two required for digital I/O, one for counter/timer/pulser).
1 PCI-20057T-1 Analog Termination Panel.
1 PCI-20058T-1 Digital Termintion Panel.

RECOMMENDED CONFIGURATION "E" OPTIONS

1 PCI-20046S-1 Software Support Package for Microsoft/IBM BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan ASYST.
1 PCI-20029A-1 Quad Termination Panel Enclosures.
1 PCI-20028A-2 Strain Relief Bracket.

If your application requires isolation of digital signals, you could use three PCI-20048T-1 Opto-isolation Termination Panels instead of the PCI-20058T-1 Digital Termination Panels. These panels accept the PCI-1100 Series digital opto-isolator modules. A PCI-20018T-1 Opto-isolation Termination Panel could be substituted which accepts up to eight isolators.

One PCI-20029A-1 Quad Termination Panel Enclosure can house up to four analog or digital termination panels.

The PCI-20028A-2 Strain Relief Bracket is used to securely fasten the input/output cables at the rear of the personal computer.

CONFIGURATION "F"—6 CHANNELS OF HIGH-RESOLUTION ANALOG OUTPUT, PLUS AN OPTIONAL 32 POINTS OF DIGITAL INPUT/OUTPUT

This configuration features six channels of 16-bit analog voltage output. A multimodule cable and a termination panel are included. If all six channels of analog output are not required, one or two PCI-20006M-2 Analog Output Modules may be eliminated or different modules substituted. If another analog output module is substituted, such as the PCI-20003M 12-bit Analog Output Module, additional cables or termination panels are not required. An optional 32 points of digital input/output are available.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "F"

1 PCI-20001C-1 Carrier.
3 PCI-20006M-2 Dual 16-bit Analog Output Modules.
1 PCI-20032A-1 Multimodule Analog Output Cable.
1 PCI-20010T-1 Analog Termination Panel.

RECOMMENDED CONFIGURATION "F" OPTIONS

1 PCI-20046S-1 Software Support Package for Microsoft/IBM BASIC language programming —or—

PCI-20046S-2 Software Support Package for Microsoft "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan ASYST.

1 PCI-20029A-1 Quad Termination Panel Enclosures.
1 PCI-20028A-2 Strain Relief Bracket.

A PCI-20001C-2 Carrier could be substituted for the PCI-20001C-1 Carrier if 32 points of buffered TTL-level digital input/output are desired. Digital cables and termination panels would also be required.

One PCI-20029A-1 Quad Termination Panel Enclosure can house up to four analog or digital termination panels.

The PCI-20028A-2 Strain Relief Bracket is used to securely fasten the input/output cables at the rear of the personal computer.

CONFIGURATION "G"—4-CHANNEL SIMULTANEOUS SAMPLE/HOLD ANALOG DATA ACQUISITION WITH PROGRAMMABLE TRIGGERING, PLUS AN OPTIONAL 32 POINTS OF DIGITAL INPUT/OUTPUT

This configuration allows the simultaneous capture of two, three or four analog signals. The start of the capture can be controlled by a trigger module that monitors an analog signal for "less than," "greater than," "inside a window" or "outside a window" conditions. In addition to the four simultaneous capture channels, there are eight additional analog inputs available plus an optional 32 points of buffered TTL-level digital input/output.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "G"

1 PCI-20001C-1 Carrier.
1 PCI-20019M-1 Data Acquisition Module (12-bit, fast A/D converter).
1 PCI-20017M-1 Four Channel Sample/Hold Module.
1 PCI-20020M-1 Trigger/Alarm Module.
2 PCI-20012A-1 Analog Shielded Cables.
1 PCI-20057T-1 Analog Termination Panel.

RECOMMENDED CONFIGURATION "G" OPTIONS

1 PCI-20046S-1 Software Support Package for Microsoft/IBM BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan ASYST.
1 PCI-20029A-1 Quad Termination Panel Enclosures.
1 PCI-20028A-2 Strain Relief Bracket.

A PCI-20001C-2 Carrier could be substituted for the PCI-20001C-1 Carrier if 32 points of buffered TTL-level digital input/output are desired. Digital cables and termination panels would also be required.

One PCI-20029A-1 Quad Termination Panel Enclosure can house up to four analog or digital termination panels. Separate enclosures are available for the PCI-20048T-1 Panel.

The PCI-20028A-2 Strain Relief Bracket is used to securely fasten the input/output cables at the rear of the personal computer.

CONFIGURATION "H"—UP TO 32 CHANNELS, TRIGGERED ANALOG DMA INPUT

This configuration samples 32 analog channels, using DMA, at throughput speeds up to 89,000 samples per second synchronized to the system's programmable pacer clock. Data is put into a circular buffer of up to 64K bytes. A Trigger/Alarm Module is used to detect either an analog or digital event, simulating the "slope and level" triggering found on an oscilloscope. Sampling can start on computer command and stop "n" samples after the trigger. This allows the option for both pre-trigger and post-trigger data to be left in the buffer. The construction of the data acquisition and analysis program can be greatly simplified by taking advantage of the PCI-20046S and PCI-20047S Software Support Packages ("drivers").

If desired, up to 32 points of digital input, available on the carrier, can also be included in the DMA data acquisition.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "H"

1 PCI-20041C-3 High Performance Carrier
1 PCI-20019M-1 High Speed Data Acquisition Module
1 PCI-20031M-1 High Speed Analog Expander
1 PCI-20020M-1 Trigger/Alarm Module
3 PCI-20012A-1 Analog Shielded Cables
1 PCI-20057T-1 High Density Analog Termination Panel

RECOMMENDED CONFIGURATION "H" OPTIONS

1 PCI-20046S-1 Software Support Package for BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft/Lattice "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan Asyst language programming.
1 PCI-20047S-1 High-Performance Software Support Package (language independent)
1 PCI-20028A-2 Strain Relief Bracket.

CONFIGURATION "I"—UP TO 512 DIGITAL I/O POINTS WITH DMA CONTROL

This configuration is for large systems where many digital I/O points are to be monitored or controlled. Four Carriers are linked together, each with 32 digital I/O points on-board. Three Digital I/O Modules, each with an additional 32 points, are installed on each of the four Carriers. Any of the 512 points can be configured as inputs or outputs in groups of eight channels each. Each point can be scanned up to 5,000 times per second under DMA control (320,000 bytes/sec throughput). Alternately, two groups of channels can be configured, with one group under DMA control and the other under program control. Applications could include factory monitoring and control, digital pattern generation, electronic score boards and process control. The component list shown is a maximum configuration. A subset of this configuration will also work fine.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "I".

4 PCI-20041C-3 High Performance Carriers
12 PCI-20004M-1 Digital I/O Modules
11 PCI-20058T-1 High Density Digital Termination Panels
32 PCI-20061A-1 Shielded digital Cables
3 PCI-20062A-1 Inter Carrier Cables

RECOMMENDED CONFIGURATION "I" OPTIONS

1 PCI-20046S-1 Software Support Package for BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft/Lattice "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan Asyst language programming.
1 PCI-20047S-1 High Performance Software Support Package (language independent)
4 PCI-20028A-2 Strain Relief Brackets.

CONFIGURATION "J"—4 CHANNEL CONTINUOUS WAVEFORM OUTPUT WITH OPTIONAL ANALOG INPUT

This configuration provides 4 continuous analog output waveforms under DMA control. At 100 points per cycle, the output frequency for each waveform can be up to 1.2 KHz (corresponding to a DMA rate of 360,000K bytes/sec). The waveform for each signal is separately programmable. The addition of a Data Acquisition Module allows analog input in a foreground program while analog output occurs in a background DMA process. Since the Carrier also includes 32 points of digital I/O, digital patterns can be generated along with the analog waveforms. The analog outputs can be either 12-bit or 16-bit resolution, depending on the analog output module chosen. More channels can be added by installing additional carriers and analog output modules.

PCI-20000 SYSTEM ELEMENTS REQUIRED FOR CONFIGURATION "J"

1 PCI-20041C-3 High Performance Carrier
2 PCI-20003M-2 2 Channel, 12 Bit Analog Output Modules —or—
2 PCI-20006M-2 2 Channel, 16 Bit Analog Output Modules
1 PCI-20057T-1 High Density Analog Termination Panel
2 PCI-20012A-1 Shielded Analog Cables —or—
1 PCI-20032A-1 Special Analog Output Cable

RECOMMENDED CONFIGURATION "J" OPTIONS

1 PCI-20046S-1 Software Support Package for BASIC language programming —or—
PCI-20046S-2 Software Support Package for Microsoft/Lattice "C" language programming —or—
PCI-20046S-3 Software Support Package for Turbo Pascal language programming —or—
PCI-20046S-4 Software Support Package for Macmillan Asyst language programming.
1 PCI-20047S-1 High Performance Software Support Package (language independent)
1 PCI-20028A-2 Strain Relief Bracket.

BURR-BROWN®

PCI-20056K-1

Complete System

PCI ControLOGraph

A Fully Integrated Data-Logging System for IBM-Compatible Personal Computers

FEATURES

- Integrated data-logging, real-time display, alarm, control and graphic analysis package
- Data recorded in hours, minutes, seconds, and tenth seconds along with day, month, and year
- Completely menu driven, NO programming required, NO previous computer skills required
- 48 inputs—
  - –21 differential analog inputs, including thermocouples
  - –24 digital inputs
  - –3 counter/frequency inputs
- 8 digital alarm and control outputs controlled by analog, digital, and/or counter channels
- Trigger options (analog, digital, and/or counter) allow both pre- and post-trigger data to be viewed
- Auto-restart allows an unattended system to resume taking data after recovery from a power failure
- Graphics analysis incorporates "active cursors" to display a given data point or the difference between data points
- ASCII data files are compatible with Lotus, ASYST, BASIC, etc., providing for additional post-acquisition analysis
- System includes a PC plug-in carrier board, modules, termination panels, cables, software, and operations manual

APPLICATIONS

- General data-logging functions, replaces chart recorders and printers while providing extensive analysis capabilities
- Time studies
- Laboratory data collection and control
- Life test and burn-in operations
- Utilities monitoring

DESCRIPTION

The PCI ControLOGraph System is a fully integrated, IBM PC-compatible system of hardware and software for data logging, real-time display, alarm annunciation, digital control, and graphics analysis. The system is completely menu-driven and requires no programming or previous experience in the use of personal computers. The system includes all necessary PCI hardware and software. The hardware is factory configured for immediate use after installation in the PC. Almost any IBM PC, PC/XT, PC/AT or compatible computer is suitable. Provisions for the easy connection of the user's sensing and control devices are provided through screw terminal panels.

SYSTEMS PRODUCTS PCI-20056K-1 DATA SHEET

10

Sensor inputs can include 21 analog signals, 24 discrete levels and 3 frequency/pulse sources. All 48 channels constitute a "frame" of data which is recorded, as a group, upon command. Each channel can be labeled and referred to by any unique, user-defined name. On each analog channel, one of four input ranges, covering $\pm$10mV to $\pm$10V, can be menu-selected. When using thermocouples, any of the channels can be designated as J, K or T type devices. Each of the counter inputs can be used to measure frequency or count events. Data can be recorded and displayed in any units desired. The user can define both linear and arbitrary nonlinear relationships between input signals and recorded/displayed data. The digital channels can monitor the state of individual bits or any combination of all 24.

The data collection process is safeguarded by two unique features that help insure that available data cannot be lost. All channels can be recorded directly to nonvolatile disk media. This means that in the event of an AC power interruption, all data is permanently saved. In addition, the system's auto-restart feature automatically "reboots" the computer and resumes the data-logging function when power is restored. In conjunction with the computer's real-time clock, the ControLOGraph provides accurate time stamping of the data—both before power interruption and after power restoration. Acquired data is continuously stored to disk in a circular buffer. The size and characteristics of the buffer can be tailored to fit the job's requirements. The circular buffer architecture provides several important features, including the capture of pre-trigger data. Up to 30,000 frames (48 input channels each) of data can be stored.

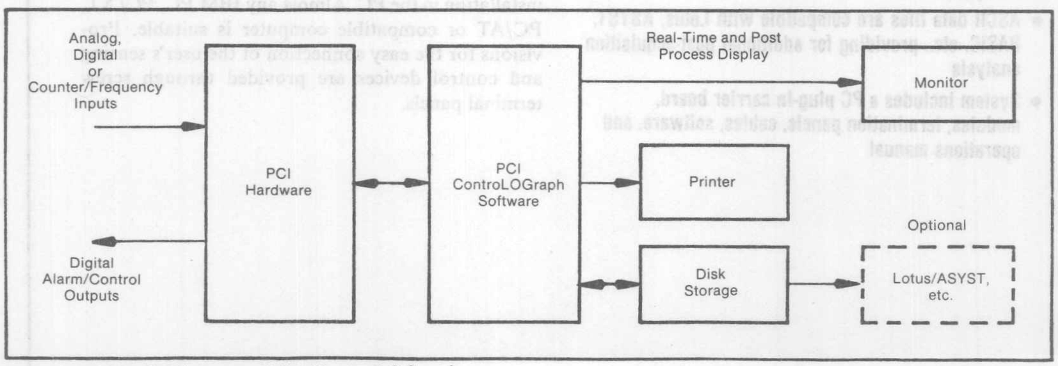
SOFTWARE

The user selects from clearly described menu options in order to acquire analog, digital, and counter data. Arbitrary functions can be defined to linearize or translate the input data into desired engineering units. Digital alarms or control outputs can be keyed to any combination of input signals. Outputs can be generated when the inputs are below, above, within a window or outside a window as defined by preset limits. Comprehensive triggering capabilities allow unique events (including fault conditions) to capture data both before and after the trigger. Numerous options allow the ControLOGraph to be customized for specific uses. System setups can be saved for later recall, enabling the users to develop a library of applications.

Data acquisition rate can be set in two ways—internal and external. The internal mode establishes a fixed timebase that can be set in 1/10-second increments, with intervals up to almost 100 hours. Each time "tick" causes a frame of data to be read. The external mode acquires a frame of data for each TTL input pulse. In both cases the data is time stamped and saved under a user-defined file name. After the acquisition is complete, the data can be recalled from disk for display and/or analysis. Built-in graphical and tabular displays are included. In the graphics mode any one, four, or as many as eight channels (digital only) can be viewed at the same time. "Zoom" capabilities allow detailed inspection of any region of the overall data set.

Also included are "find" functions. The find functions can be used to search for specific data values, the maxima or minima of the data set and digital events or patterns. Auto-scaling provides maximum visual resolution, while the digital display of the cursor's time and amplitude position yields precise data readings. Data corresponding to the cursor position can be presented in several modes: real time, date, trigger relative, and origin relative. The real time and date modes yield the actual time of the data in an hh:mm:ss.s and day/month/year format, respectively. Data can also be presented relative to the system trigger or any selected point. This last mode provides precise "delta" readings.

A special file building function generates ASCII format files that can be read by Lotus, ASYST, or DADiSP to provide additional data reduction, analysis or presentation capabilities.



Functional Block Diagram of the ControLOGraph.

HARDWARE REQUIREMENTS

DATA ACQUISITION HARDWARE REQUIREMENTS

Hardware for the PCI ControLOGraph System is from the PCI-20000 family of intelligent instrumentation products. A complete system includes the following hardware items:

1	PCI-20001C-2	Carrier with digital I/O
1	PCI-20002M-1	Analog-input instrument module
1	PCI-20005M-1	Analog-input expansion instrument module
1	PCI-20007M-1	Counter/timer instrument module
1	PCI-20057T-1	High-density analog-input termination panel
3	PCI-20012A-1	6-foot (2m) shielded analog cable
1	PCI-20058T-1	High density digital I/O termination panel
3	PCI-20061A-1	6-foot (2m) ground plane digital cable
1	PCI-20028A-2	Strain-relief bracket for cables

PERSONAL COMPUTER SYSTEM REQUIREMENTS

The PCI ControLOGraph system is designed to work with an IBM PC, PC/XT, PC/AT or any other compatible personal computer. Minimum computer requirements are:

- 256k random access memory (RAM)
- Two double-sided, double-density, floppy disk drives, or one DSDD floppy drive and a hard disk drive
- IBM compatible color graphics adapter card (CGA), or extended graphics adapter (EGA) card, and a compatible color or monochrome monitor.
- PC-DOS or MS-DOS, version 2.0 or higher.

SYSTEM SPECIFICATIONS

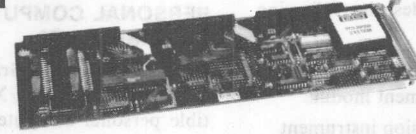
PCI-20056K-1 ControLOGraph

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	SPECIFICATION
INPUT FRAME (total channels)	48 channels
Analog (differential input, 12-bit resolution)	21 channels
Voltage ranges	±10mV/±10V
Thermocouple types	J, K, T
Digital (TTL)	24 channels
Counter/Frequency	3 channels
Maximum count	65,535
Maximum frequency	650kHz
OUTPUT CHANNELS	
Digital Alarm/Control	8 channels
Output levels	TTL
Sink/source current	24mA/15mA
Control stimulus	Analog, digital, counter
SAMPLE RATE AND PERIOD (IBM PC/AT)	
To RAM Disk (channels/second and seconds/frame)	480 and 0.1
To Hard Disk (channels/second and seconds/frame)	80 and 0.6
To Floppy Disk (channels/second and seconds/frame)	12 and 4
SAMPLE RATE AND PERIOD (IBM PC/XT)	
To RAM Disk (channels/second and seconds/frame)	480 and 0.1
To Hard Disk (channels/second and seconds/frame)	24 and 2
To Floppy Disk (channels/second and seconds/frame)	8 and 6
FRAME ACQUISITION TIME (max)	0.1s
DATA BUFFER, Type	Circular
Max Size (frames)	30,000
Max Delay (see trigger types)	30,000
TRIGGER INPUT TYPES Stops acquisition after delay	Analog, Digital and Counter/Frequency
REAL-TIME DISPLAY, 16 equal-sized windows	
Can be modified during run without interruption	Any 16 Channels
GRAPHICS DISPLAY, choice of any 1 or 4 channels (8 if digital)	
Cursor-Controlled Read-Out of Data	1, 4 or 8
OUTPUT FILE COMPATIBILITY	
Can be read by Lotus, ASYST, BASIC, etc.	ASCII Real

BURR-BROWN®

PCI-20001C SERIES



GENERAL-PURPOSE CARRIER BOARD

FEATURES

- PLUGS INTO INTERNAL EXPANSION SLOT OF ANY IBM COMPATIBLE COMPUTER: IBM PC, XT, AT; COMPAQ; AT&T; ZENITH; ETC.
- FUNCTIONALITY PROGRAMMED VIA A FAMILY OF PLUG-IN INSTRUMENT MODULES
- PROVISIONS FOR UP TO THREE INSTRUMENT MODULES PER CARRIER
- NO EXTERNAL POWER REQUIRED
- ON BOARD BUS ALLOWS DIGITAL, ANALOG AND TIMING SIGNALS TO BE PASSED BETWEEN MODULES
- MEMORY MAPPED ADDRESSING ALLOWS A LARGE NUMBER OF CARRIERS TO BE CONNECTED TO ONE PC
- A CARRIER IS AVAILABLE WITH 32 DIGITAL I/O POINTS INSTALLED. THIS CAPABILITY LEAVES ALL THREE MODULE LOCATIONS FREE FOR FURTHER EXPANSION.

DESCRIPTION

The PCI-20001C-1 "basic" carrier is designed to interface directly with the IBM PC's internal bus through any available expansion slot. Major features of this carrier include the Intelligent Instrumentation Interface (I^3) bus and its capacity for up to three Instrument Modules. Please refer to Figures 1 and 4. The I^3 bus supports digital, sync and analog signal communication. The sync lines make possible the coordination (timing) of various system elements. The differential analog chain allows any module to condition its input signal and then pass the result to the next module. Bus translation circuitry links the IBM computer bus to the I^3 bus. Logic for interrupt control, carrier identification and module selection is also included. All power is derived from the +5V, PC power supply. A DC/DC converter on the carrier generates ±15V for use by the modules.

Several carriers can be installed in one PC, up to the limits on available expansion slots. However, the mechanical thickness of the carrier/module assembly, and power requirements can limit a practical installation to less than the number of slots. Each carrier is addressed into the memory map of the PC, and requires one kilobyte of space. DIP switches allow placing the carrier anywhere within the one megabyte of available memory address space.

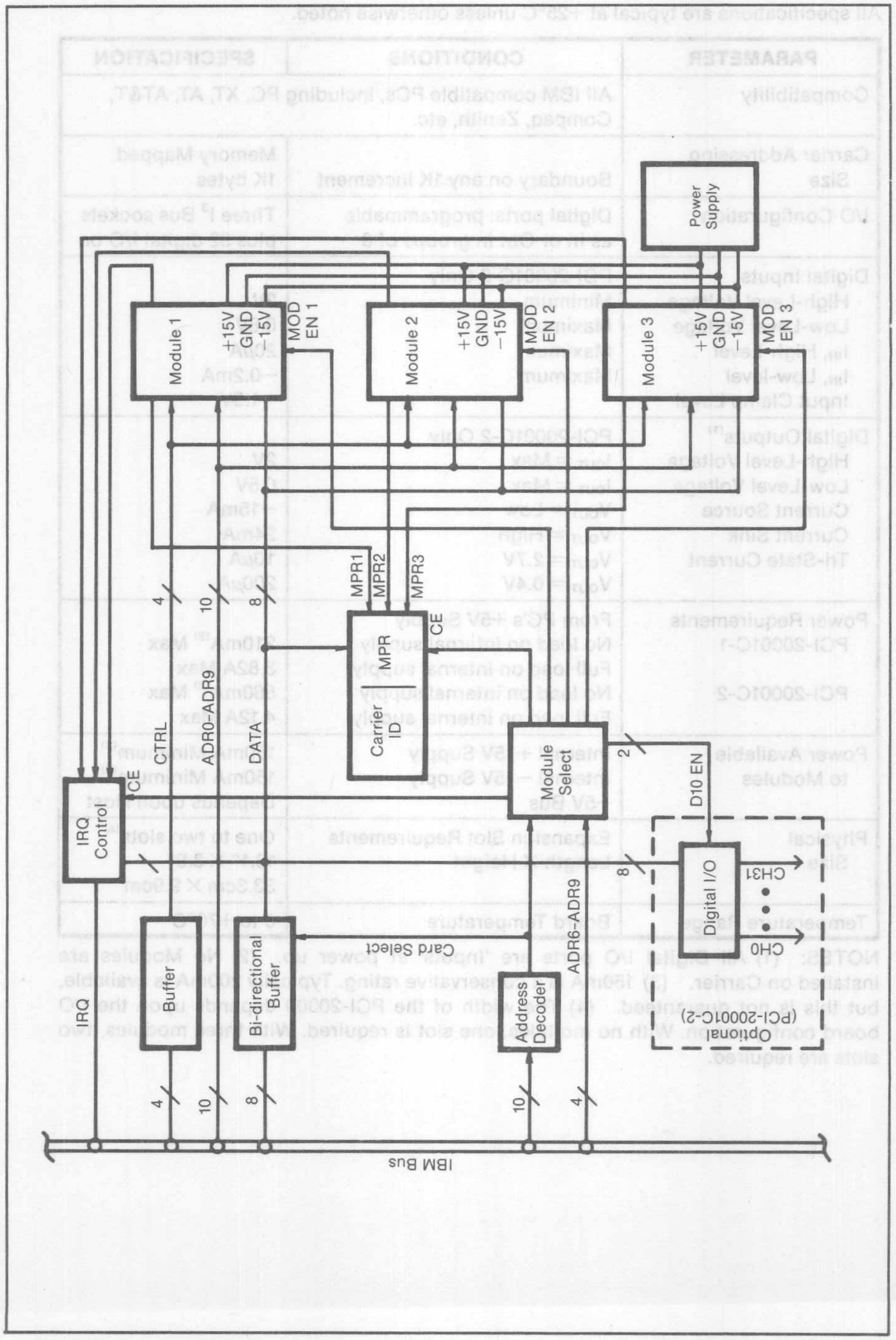
The PCI-20001C-2 carrier possesses all of the physical and electrical characteristics of the PCI-20001C-1 plus one additional feature. The PCI-20001C-2 has 32 points of fully buffered digital I/O. (See block diagram.) The 32 points are arranged in four groups of eight bits (bytes). Each byte can, under software control, be configured for either input or output use. Field connections to these I/O points are made through two connectors on the carrier. Each connector supports two bytes. Ribbon cables are used to interconnect the carrier and the appropriate signal termination panels. This digital I/O capacity does not diminish any of the other functions, and leaves all three instrumentation module positions free for further expansion.

SPECIFICATIONS—PCI-20001C-1, PCI-20001C-2

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility	All IBM compatible PCs, including PC, XT, AT, AT&T, Compaq, Zenith, etc.	
Carrier Addressing Size	 Boundary on any 1K increment	Memory Mapped 1K bytes
I/O Configuration	Digital ports: programmable as In or Out in groups of 8	Three I^3 Bus sockets plus 32 digital I/O on
Digital Inputs High-Level Voltage Low-Level Voltage I_{IN}, High-Level I_{IN}, Low-level Input Clamp Level	PCI-20001C-2 Only Minimum Maximum Maximum Maximum	 2V 0.8V 20μA −0.2mA −1.5V
Digital Outputs[1] High-Level Voltage Low-Level Voltage Current Source Current Sink Tri-State Current	PCI-20001C-2 Only I_{OUT} = Max I_{OUT} = Max V_{OUT} = Low V_{OUT} = High V_{OUT} = 2.7V V_{OUT} = 0.4V	 2V 0.5V −15mA 24mA 10μA 200μA
Power Requirements PCI-20001C-1 PCI-20001C-2	From PC's +5V Supply No load on internal supply Full load on internal supply No load on internal supply Full load on internal supply	 210mA[2] Max 3.82A Max 560mA[2] Max 4.12A Max
Power Available to Modules	Internal +15V Supply Internal −15V Supply +5V Bus	150mA Minimum[3] 150mA Minimum[3] Depends upon Host
Physical Size	Expansion Slot Requirements Length × Height	One to two slots[4] 13.1″ × 3.9″ 33.3cm × 9.9cm
Temperature Range	Board Temperature	0 to +70°C

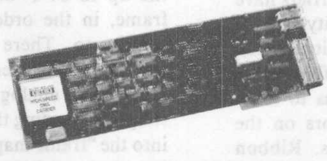
NOTES: (1) All Digital I/O ports are 'Inputs' at power up. (2) No Modules are installed on Carrier. (3) 150mA is a conservative rating. Typically 200mA is available, but this is not guaranteed. (4) The width of the PCI-20000 depends upon the I/O board configuration. With no modules, one slot is required. With three modules, two slots are required.



PCI-20001C Carrier Block Diagram.

BURR-BROWN®

PCI-20041C SERIES



HIGH-PERFORMANCE CARRIER BOARD

FEATURES

- UNIQUE DMA TECHNIQUE SUPPORTS HIGH SPEED TRANSFERS OF ANALOG, DIGITAL AND COUNTER DATA SIMULTANEOUSLY ON UP TO 5 CARRIERS
- 360K BYTES/SEC DMA TRANSFERS
- PRE-TRIGGER AND POST-TRIGGER VIEWING OF EVENT DATA
- INTER-CARRIER BUS ALLOWS DATA TRANSFERS BETWEEN CARRIERS
- DMA, INTERRUPT DRIVEN OR POLLED MODES OF OPERATION
- PLUGS INTO EXPANSION SLOT OF IBM PC COMPATIBLE COMPUTERS
- FUNCTIONALITY DETERMINED BY UP TO THREE PLUG-IN INSTRUMENT MODULES
- ON-BOARD BUS ALLOWS DIGITAL, ANALOG AND TIMING SIGNALS TO PASS BETWEEN MODULES

DESCRIPTION

The PCI-20041C-2 and PCI-20041C-3 (DMA version) are carrier boards which interface directly with the internal bus of the IBM PC or compatible computers through any available expansion slot. Each carrier provides mounting space, power, and intermodule communications for up to three instrument modules of the PCI-20000 Data Acquisition, Test, Measurement, and Control System. Digital, sync, and analog signals may be passed between modules via the on-board Intelligent Instrumentation Interface (I^3) bus. In addition, an inter-carrier communications port allows similar communications with up to 15 modules residing on multiple carriers. This allows the building of systems ranging from those with a small number of I/O points and relative simplicity to those of several hundred I/O channels, high speed, and considerable sophistication.

The sync lines of the I^3 bus make possible the coordination of multiple instrument modules to perform various signal processing functions. The differential analog chain and the digital lines of this bus permit any module to condition its input signal and to pass the result to the next module. Bus translation circuitry on the carrier links the IBM PC

bus to the I^3 bus. Logic for interrupt control, carrier identification and module selection is also included. All power is derived form the +5VDC power supply of the PC. A DC/DC converter on the carrier generates regulated ±15VDC power for use by the instrument modules.

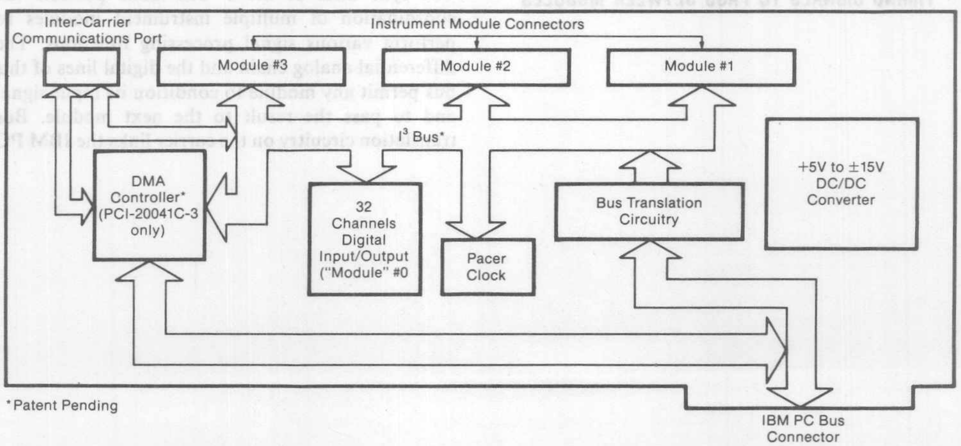
Both the PCI-20041C-2 and PCI-20041C-3 Carriers have 32 points of fully buffered digital I/O capability. The 32 points are arranged in four groups of eight bits (bytes). Each byte can, under software control, be configured for either input or output use. Field connections to these I/O points are made through two connectors on the carrier. Each connector supports two bytes. Ribbon cables are used to interconnect the carrier to appropriate signal termination panels. This digital I/O capacity does not diminish any of the other functions, and it leaves all three instrument ,module positions free for further expansion.

Both carrier models also include an 8MHz programmable pãcer clock for use in the timing of data acquisition and transfers of data to and from memory. Both the PCI-20041C-2 and PCI-20041C-3 can operate in the programmed transfer mode using either "polling" or "interrupt" techniques. Each carrier supports a single interrupt to the host computer. The interrupt can be jumpered to levels 3 through 7.

In addition to all of the above capabilites, a unique (patent pending) DMA technique gives the PCI-20041C-3 Carrier the ability to make very high speed transfers (up to 360K bytes/sec) of input (or output) data to or from the memory of the PC. The DMA controller in the host computer is used in conjunction with a controller on the carrier to accomplish the transfer invisibly to the host computer. Data is transferred to or from the memory of the host PC in frames of up to 64 bytes using DMA. Block transfers of up to 64K bytes can be stored in RAM. For input transfers, frames of data from input modules on the I^3 bus are stored to sequential locations in the host computer's memory. For output transfers, frames of data from specified sequential locations in the host computer's memory are transferred to specified output modules on the I^3 bus. The "frame map" is stored in a block of memory on the carrier itself. This is a list of the up to 64 I^3 bus addresses which are to be in the frame, in the order in which they are to be read or written to. There is no need for this list of I^3 bus addresses to be sequential. This allows the high speed scanning of analog or digital inputs in any desired order simply by loading the desired sequence of I^3 bus addresses into the "frame map" memory. Conversely, output DMA transfers may be made to any desired sequence of output devices on the I^3 bus simply by specifying the appropriate "frame map."

The inter-carrier communications port allows a designated carrier to operate as a "master" and to control DMA transfers from up to four other carriers which operate as "slaves." This is done by insertion of the appropriate jumpers on each carrier and by connecting the carriers together via inter-carrier ribbon cables. By chaining carriers together in this way, it is possible to make DMA transfers to or from up to 15 instrument modules residing on the carriers. The maximum frame size is still 64 bytes. The sequence of module addresses is once again arbitrarily determined by the user to meet the requirements of his application.

DMA transfers may be initiated on command or on the occurrence of an event after delay. Transfers may be terminated on command, on the occurrence of an event after delay, or after transferring a specified number of frames of data. Transfers can be timed by the on-board pacer clock, or by the occurrence of events. Transfers of data to a circular buffer can be used to give pre- and post-trigger information. This allows the analysis of conditions both before and after the occurrence of a random critical event.



*Patent Pending

SPECIFICATIONS—PCI-20041C-2A, PCI-20041C-3A

All specifications are typical at +25°C unless otherwise noted.

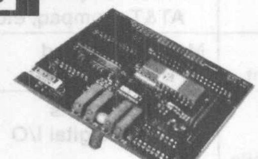
PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All IBM compatible PCs, including PC, XT, AT, AT&T, Compaq, etc.
Carrier Addressing		Memory Mapped
Size	Boundary on any 1K increment	1K Bytes
I/O Configuration		3 I^3 Bus sockets
	Digital ports: programmable as In or Out in groups of 8 bits	plus 32 digital I/O
Digital Inputs		
High-Level Voltage	Minimum	2V
Low-Level Voltage	Maximum	0.8V
I_{IN}, High-Level	Maximum	70μA
I_{IN}, Low-Level	Maximum	−0.25mA
Input Clamp Level		−1.5V
Digital Outputs[1]		
High-Level Voltage	I_{OUT} = MAX	2.4V
Low-Level Voltage	I_{OUT} = MAX	0.5V
Current Source	V_{OUT} = Low	−6.5mA
Current Sink	V_{OUT} = High	24mA
Tri-State Current	V_{OUT} = 2.7V	70μA
Tri-State Current	V_{OUT} = 0.4V	250μA
Pacer Clock	Basic Frequency	8MHz ±.01%
Output Frequency	N_1 and N_2 are 16-bit integers	8MHz/($N_1 \cdot N_2$)
Interrupts	Can be Latched	
Levels	Jumper Selectable	2 through 7
Sources	Via IRQ0*	Modules 1-3, Pacer Clock, External TTL
Sense		TTL high to low
DMA Transfers	PCI-20041C-3A Only	
Data Types	Analog, Digital, Counter	Inputs or Outputs
Speed	Maximum Rate[2]	360K Bytes/second
Transfer Modes	Linear or Circular Buffers	Start and Stop on Command or on Event After Delay
Frame Size	Maximum number of addresses	64 Bytes
Block Size	Maximum data stored in RAM	64K Bytes
Power Requirements	From PC's +5V Supply	
PCI-20041C-2	No Load on Internal Supply	1.7A[3] Typical, 2.7A[3] Maximum
	Full Load on Internal Supply	3.85A Typical, 4.85A Maximum
PCI-20041C-3A	No Load on Internal Supply	2.5A[3] Typical, 3.6A[3] Maximum
	Full Load on Internal Supply	4.65A Typical, 5.75A Maximum
Power Available to Modules	Internal +15V Supply	150mA Minimum[4]
	Internal −15V Supply	150mA Minimum[4]
	+5V Bus	Depends Upon Host
Physical	Expansion Slot Requirements	1 to 2 Slots[5]
Size	Length × Height	13.1" × 3.9", 33.3cm × 9.9cm
Temperature Range	Board Temperature	0 to 70°C

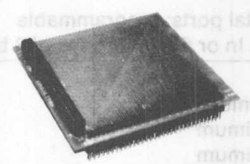
NOTES: (1) All Digital I/O ports are "Inputs" at power up. (2) The obtainable DMA transfer rate depends upon several factors including the PC type, frame size, mode, etc. (3) No Modules are installed on Carrier. (4) 150mA is a conservative rating. Typically 200mA is available, but this is not guaranteed. (5) The width of the PCI-20000 depends upon the I/O board configuration. With no modules, one slot is required. With three modules, two slots are required.

BURR-BROWN®

PCI-20002M-1

Analog Input Module





PCI-20033A-1

Extender Board

ANALOG INPUT MODULE
for General-Purpose Use and for Low-Level Inputs

FEATURES

- COMPATIBLE WITH I³ BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT
- 16 SINGLE-ENDED OR 8 DIFFERENTIAL INPUTS
- 12-BIT RESOLUTION
- 0.04% LINEARITY ERROR
- UP TO 32kHz SAMPLE RATE

DESCRIPTION

This module accepts a wide range of analog input signals and performs the A/D conversions necessary to make the data compatible with digital computers. Below is a functional block diagram of the module. Input multiplexers select any one of 16 single-ended input channels. Alternatively, the mux can be jumper programmed for eight differential channels. Additional input channels can be obtained by using the optional PCI-20005M-1 Expansion Module. Each Expansion Module adds 32 single-ended (16 differential) channels. A high performance, differential input, programmable gain amplifier provides signal scaling and common-mode rejection. Gains of 1, 10, 100 and 1000 are available under software control. The 12-bit A/D converter can be set up for an input range of ±5V, 0 to 10V or ±10V full scale. Input signals are usually connected to external termination panels, and brought to the module via shielded ribbon cable. The PCI-20033A-1 is a module extender board designed to allow easy access to the calibration potentiometers on the PCI-20002M-1 module. The extender board fits between the connectors on the carrier on the module.

SPECIFICATIONS—PCI-20002M-1

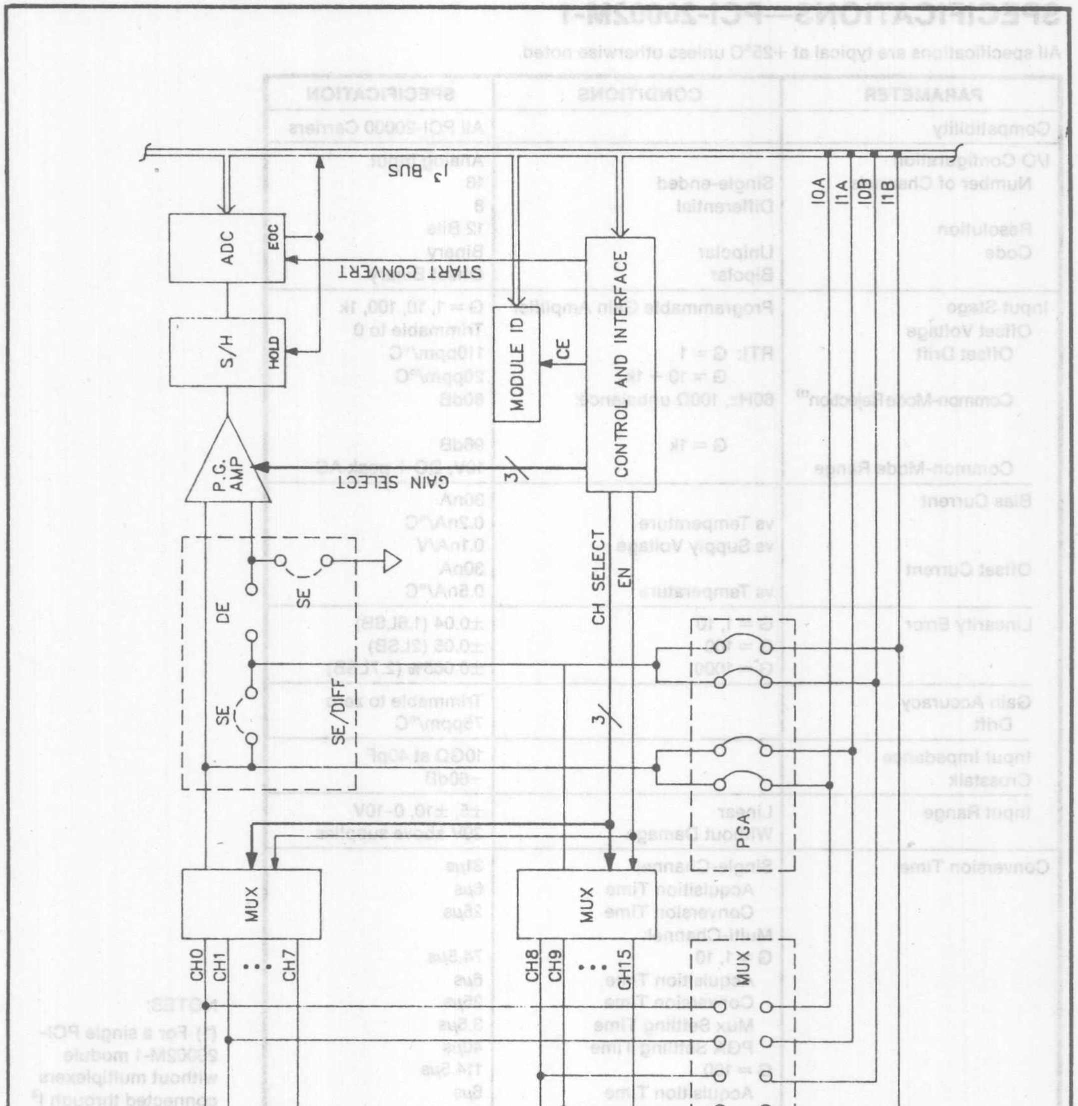
All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
I/O Configuration		Analog Input
Number of Channels	Single-ended	16
	Differential	8
Resolution		12 Bits
Code	Unipolar	Binary
	Bipolar	Offset Binary
Input Stage	Programmable Gain Amplifier	G = 1, 10, 100, 1k
Offset Voltage		Trimmable to 0
Offset Drift	RTI: G = 1	110ppm/°C
	G = 10 − 1k	20ppm/°C
Common-Mode Rejection(1)	60Hz, 100Ω unbalance:	80dB
	G = 1k	96dB
Common-Mode Range		10V, DC + peak AC
Bias Current		30nA
	vs Temperature	0.2nA/°C
	vs Supply Voltage	0.1nA/V
Offset Current		30nA
	vs Temperature	0.5nA/°C
Linearity Error	G = 1, 10	±0.04 (1.6LSB)
	G = 100	±0.05 (2LSB)
	G = 1000	±0.065% (2.7LSB)
Gain Accuracy		Trimmable to zero
Drift		75ppm/°C
Input Impedance		10GΩ at 40pF
Crosstalk		−60dB
Input Range	Linear	±5, ±10, 0–10V
	Without Damage	20V above supplies
Conversion Time	Single-Channel	31μs
	Acquisition Time	6μs
	Conversion Time	25μs
	Multi-Channel:	
	G = 1, 10	74.5μs
	Acquisition Time	6μs
	Conversion Time	25μs
	Mux Settling Time	3.5μs
	PGA Settling Time	40μs
	G = 100	114.5μs
	Acquisition Time	6μs
	Conversion Time	25μs
	Mux Settling Time	3.5μs
	PGA Settling Time	80μs
	G = 1000	704.5μs
	Acquisition Time	6μs
	Conversion Time	25μs
	Mux Settling Time	3.5μs
	PGA Settling Time	650μs
Power Requirements(2)	+15V Supply	32mA Max
	−15V Supply	56mA Max
	+5V Supply	200mA Max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3" 9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

NOTES:

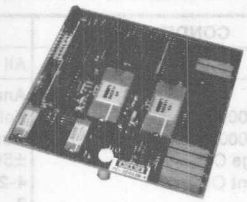
(1) For a single PCI-20002M-1 module without multiplexers connected through I^3 Bus. Connections to the Bus can reduce CMR.

(2) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter, and the equivalent load on the computer's +5V supply will be 730mA, maximum. This takes into account the efficiency of the DC/DC converter.



PCI-20002M Module Block Diagram.

BURR-BROWN®



PCI-20003M-2
PCI-20003M-4

12-BIT ANALOG OUTPUT MODULES

FEATURES

- COMPATIBLE WITH I³ BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT
- 12-BIT RESOLUTION
- ±1/2LSB LINEARITY ERROR
- 3μs SETTLING TIME
- VOLTAGE AND CURRENT OUTPUTS AVAILABLE

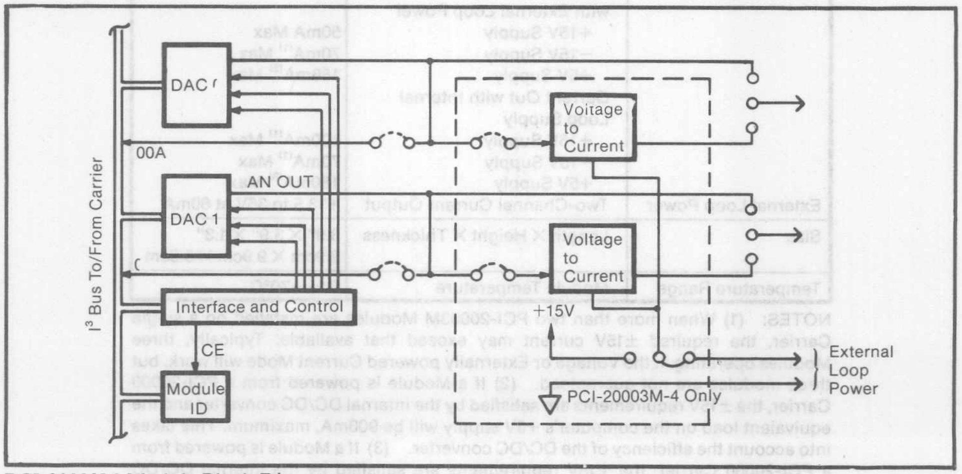
DESCRIPTION

The PCI-20003M-2 module accepts digital code from a computer and generates analog output voltages in the range of ±10V. In addition to the voltage outputs, the PCI-20003M-4 also has 4 to 20mA current outputs available. Below is a functional block diagram of the PCI-20003M-4. Both the PCI-20003M-2 and -4 modules contain 2 output channels with digital-to-analog converters (DACs).

All DACs have 12-bit resolution and can be jumper programmed for ±5V, 0 to 10V and ±10V full scale output. In addition, the current output models can be jumper programmed for either 4 to 20mA or 5 to 25mA.

As is the case with all instrument modules, the I³ bus can be used to chain the output of these modules to the next module.

Output signals are usually connected to an external termination panel via shielded ribbon cable, where field connections can be easily accommodated.



PCI-20003M Module Block Diagram.

PCI-20000 INSTRUMENT MODULES
PCI-20003M SERIES DATA SHEET

10

SPECIFICATIONS—PCI-20003M-2, PCI-20003M-4

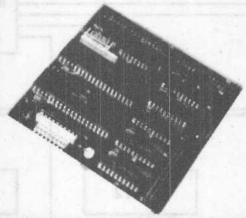
All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
Configuration		Analog Output
	PCI-20003M-2	Voltage Only
	PCI-20003M-4	Current or Voltage
Range	Voltage Output	±5V, ±10V, 0-10V
	Current Output	4-20mA, 5-25mA
Number of Channels		2
Resolution		12 Bits
Code	Unipolar	Binary
	Bipolar	Offset Binary
Linearity Error	Voltage Output, Maximum	±0.5LSB
	Current Output	±1.5LSB
Differential	Voltage Output, Maximum	±0.75LSB
Drift	0 to 50°C, Maximum	±0.75LSB
Monotonicity	0 to +50°C	Fully monotonic
Gain Accuracy	Voltage Output	Trimmable to zero
	Current Output	0.6%FSR
Drift	Voltage Output	±30ppm/°C
	Current Output	±80ppm/°C
Offset	Voltage Output	Trimmable to zero
	Current Output	0.4%FSR
Drift	Voltage Output	±10ppm/°C
	Current Output	±60ppm/°C
Output Stage		
Current	Voltage Output	±5mA
Impedance	Voltage Output at DC	0.2Ω
Compliance	Current Output	15V or Loop supply
Settling Time	Voltage Output, within 0.01%	
	20V step	4μs
	10V step	3μs
	Current Output, within 0.1%	18μs
Slew Rate	Voltage Output	8V/μs
	Current Output	1mA/μs
Conversion Rate	See Speed Summary Table	
Power Requirements	Voltage Out or Current Out with External Loop Power	
	+15V Supply	50mA Max
	−15V Supply	70mA(1) Max
	+5V Supply	180mA(2) Max
	Current Out with Internal Loop Supply	
	+15V Supply	100mA(1) Max
	−15V Supply	70mA(1) Max
	+5V Supply	180mA(3) Max
External Loop Power	Two-Channel Current Output	+13.5 to 35V at 60mA
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3" 9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

NOTES: (1) When more than two PCI-20003M Modules are installed on a single Carrier, the required ±15V current may exceed that available. Typically, three Modules operating in the Voltage or Externally powered Current Mode will work, but three modules are not guaranteed. (2) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 900mA, maximum. This takes into account the efficiency of the DC/DC converter. (3) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 1200mA, maximum. This takes into account the efficiency of the DC/DC converter.

BURR-BROWN®

PCI-20004M-1



DIGITAL INPUT/OUTPUT MODULE

FEATURES

- COMPATIBLE WITH I³ BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT
- 32 DIGITAL INPUT/OUTPUT POINTS
- TTL COMPATIBLE LEVELS
- BUFFERED OUTPUTS BOTH SOURCE AND SINK CURRENT
- DIRECTLY COMPATIBLE WITH INDUSTRY STANDARD OPTO-ISOLATORS

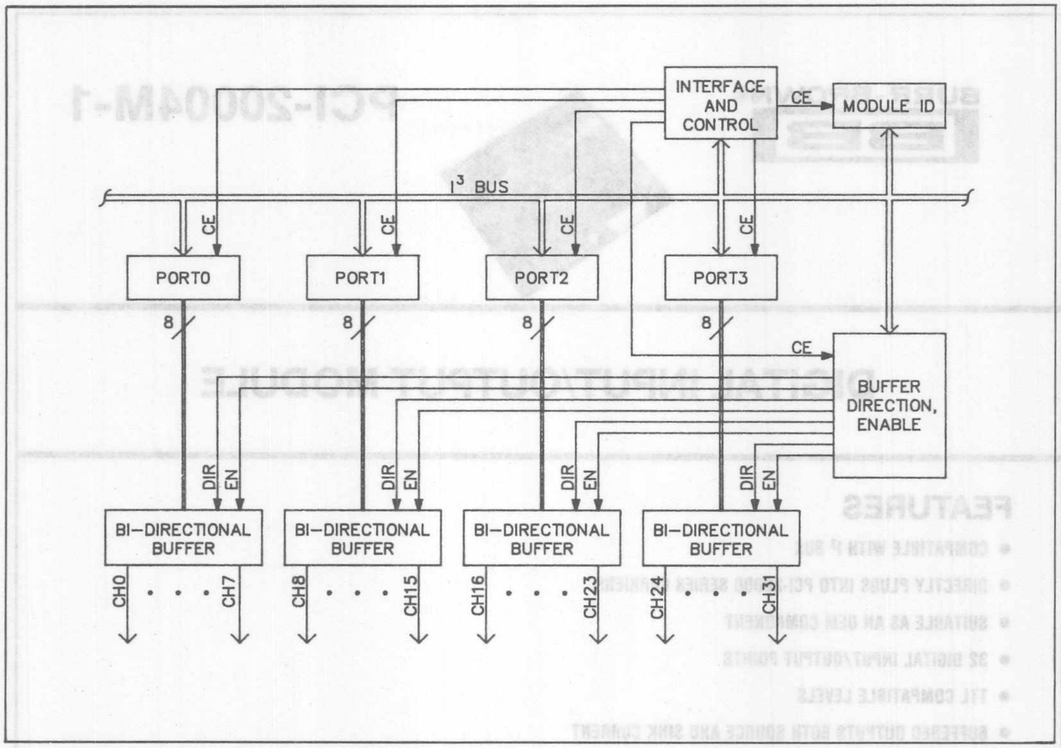
DESCRIPTION

This 32 point module functions with TTL compatible digital signals. The 32 points are arranged in 4 bytes of 8 bits each. Each byte can, under software control, be selected for either input or output use. All lines are buffered to give full, bipolar, TTL drive capability. A block diagram is shown below.

The module can monitor or control devices having discrete on/off states such as relays, switches, lamps, etc. Through the use of optical couplers (PCI-1100 series), non-TTL signals can also be interfaced. For example, loads like AC or DC motors can be readily switched and AC line voltage can be detected.

In addition to reading (or writing) bytes, software can extract individual bits or assemble words. In this way, logical combinations can be tested to determine alarm or control conditions.

The field, I/O, signals are usually connected to external termination panels, and brought to the module via ground-plane ribbon cable. Both conventional, and opto-isolated termination panels are suitable for use with all PCI-20000 digital signals.



PCI-20004M-1—Digital I/O, Functional Block Diagram.

SPECIFICATIONS—PCI-20004M-1

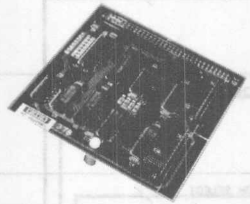
All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
I/O Configuration	Software programmable as Inputs or Outputs by bytes	32 D I/O channels
Digital Inputs:		
High-Level Voltage	Minimum	2V
Low-Level Voltage	Maximum	0.8V
I_{IN}, High-Level	Maximum	20μA
I_{IN}, Low-Level	Maximum	−0.2mA
Input Clamp Level		−1.5V
Digital Outputs(1):		
High-Level Voltage	I_{OUT} = Max	2V
Low-Level Voltage	I_{OUT} = Max	0.5V
Current Source	V_{OUT} = Low	−15mA
Current Sink	V_{OUT} = High	24mA
Tri-State Current	V_{OUT} = 2.7V	10μA
	V_{OUT} = 0.4V	200μA
Power Requirements	+5V Supply	350mA max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3" 9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

NOTES: (1) All digital I/O ports are 'inputs' at power-up.

BURR-BROWN®

PCI-20005M-1



ANALOG INPUT EXPANSION MODULE

FEATURES

- COMPATIBLE WITH I³ BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT
- ADDS 32 (SE) OR 16 (DIFF) CHANNELS TO EXISTING ANALOG INPUT COUNT
- CAN BE USED AS A GENERAL PURPOSE MULTIPLEXER
- INPUTS PROTECTED TO ±35V

DESCRIPTION

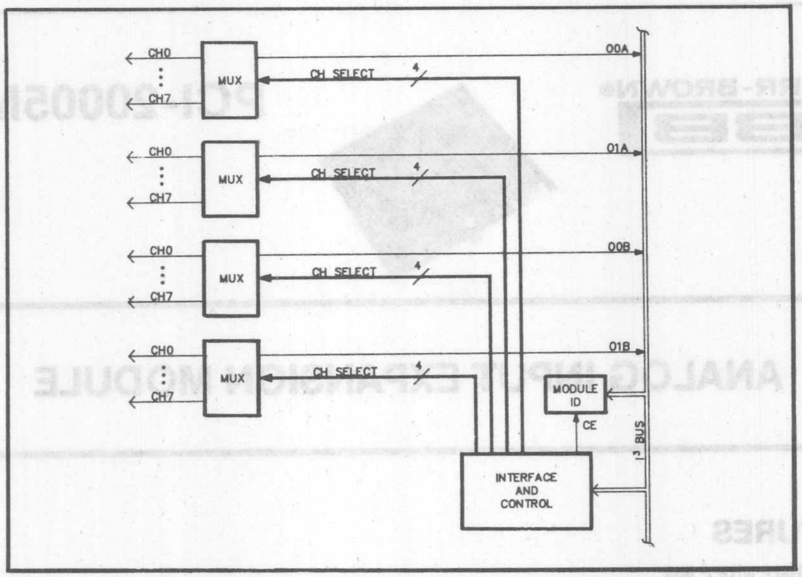
This expansion module is useful in applications that require more analog input channels than can be supported by the PCI-20002M-1 module alone. Please refer to the functional block diagram below.

The PCI-20005M-1 consists primarily of software-controlled multiplexers. One of 32 single-ended (SE), or one of 16 differential inputs can be selected for further processing. Configuring the module for SE or differential operation is also under software control.

The chain feature of the I³ bus makes it possible to connect the output of an expansion module directly to the PGA inputs on the analog input module. This adds additional channels without consuming any of the original analog inputs. Note that when chaining analog input and analog expansion modules together, both must be configured alike with respect to SE or differential use.

It is appropriate to consider using the expansion module for other, more general, multiplexing applications. As a system building block, one can envision custom requirements, which could even include "fan-out" functions.

Field signals are usually connected to external termination panels, and brought to the expansion module via shielded ribbon cables. Various termination-panel options exist depending on the number of channels required.



PCI-20005M-1 Analog Expansion Module, Functional Block Diagram.

SPECIFICATIONS—PCI-20005M-1

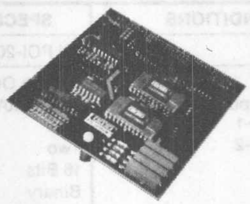
All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility	Primarily designed as an Input Multiplexer for the PCI-20002M-1 A/D Module	All PCI-20000 Carriers
I/O Configuration Number of Channels	 Single-Ended Differential	Analog Expander 32 16
Analog Signal Range	Linear Operation Without Damage	±10V 20V above supply
Input Capacitance(1)	Channel 'On': Single-Ended Differential Channel 'Off'	100pF 50pF 5pF
'On' Resistance 'Off' Isolation	Maximum Frequency = 1kHz, R_S = 1kΩ	1.8kΩ 85dB
Input Leakage	'On' Channel, at +25°C at +70°C 'Off' Channel, at +25°C at +70°C	0.1nA 2.5nA 0.03nA 0.7nA
Power Requirements(2)	+15V Supply −15V Supply +5V Supply	20mA max 8mA max 215mA max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3" 9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

NOTES: (1) For a single PCI-20005M-1 module without regard to 'loads' connected through the I^3 Bus. However, in the single-ended mode it is assumed that all four mux outputs are connected together, and in the differential mode two mux outputs are connected together. (2) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 385mA, maximum. This takes into account the efficiency of the DC/DC converter.

BURR-BROWN®

PCI-20006M-1
PCI-20006M-2



16-BIT ANALOG OUTPUT MODULES

FEATURES

- COMPATIBLE WITH I³ BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT
- 16-BIT RESOLUTION
- ±0.003% LINEARITY ERROR
- 10V/μs SLEW RATE

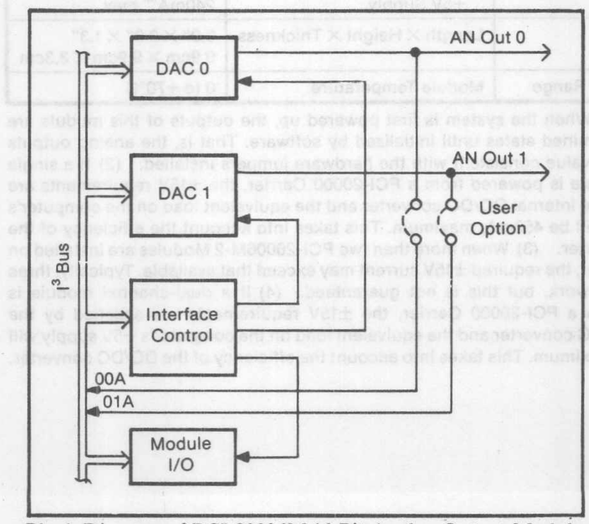
DESCRIPTION

The PCI-20006M-1 and -2 modules accept digital code from a computer and generate analog output voltages in the range of ±10V. Below is a functional block diagram of the PCI-20006M-2. The PCI-20006M-2 module contains two digital-to-analog converters (DACs). The PCI-20006M-1 has one DAC.

Both DACs have 16-bit resolution and can be jumper programmed for ±5V, 0 to 10V and ±10V full scale output.

As is the case with all instrument modules, the I³ bus can be used to chain the output of these modules to the next module.

Output signals are usually connected to an external termination panel via shielded ribbon cable, where field connections can be easily accommodated.



Block Diagram of PCI-20006M 16-Bit Analog Output Module.

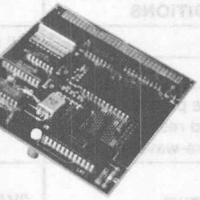
SPECIFICATIONS—PCI-20006M-1, PCI-20006M-2

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
Configuration		Analog Output
Range	Voltage Output(1)	±5V, ±10V, 0-10V
Number of Channels	PCI-20006M-1	One
	PCI-20006M-2	Two
Resolution		16 Bits
Code	Unipolar	Binary
	Bipolar	Two's Complement
Linearity Error	At +25°C	±0.002%FSR
	Over Temp. Range	±0.004%FSR
Differential	At +25°C	±0.003%FSR
	Over Temp. Range	±0.006%FSR
Monotonicity	Over Temp. Range	14 Bits
Gain Accuracy		Adjustable to Zero
Drift		±25ppm/°C
Offset		Adjustable to Zero
Drift	Unipolar	±3ppm/°C
	Bipolar	±10ppm/°C
Output Stage:		
Current		±5mA
Impedance	At DC	0.15Ω
Settling Time	To 0.003%FSR, 20kΩ Load Full Scale Step	8μs
Slew Rate		10V/μs
Conversion Rate	See Speed Summary Table	
Power Requirements	Single Channel:	
	+15V Supply	10mA Max
	−15V Supply	30mA max
	+5V Supply	210mA(2) max
	Two Channel:	
	+15V Supply	20mA Max
	−15V Supply	60mA(3) max
	+5V Supply	240mA(4) max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3"
		9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

NOTES: (1) When the system is first powered up, the outputs of this module are NOT in determined states until initialized by software. That is, the analog outputs could be any value consistent with the hardware jumpers installed. (2) If a single channel module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 450mA, maximum. This takes into account the efficiency of the DC/DC converter. (3) When more than two PCI-20006M-2 Modules are installed on a single Carrier, the required ±15V current may exceed that available. Typically, three modules will work, but this is not guaranteed. (4) If a dual-channel module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 720mA, maximum. This takes into account the efficiency of the DC/DC converter.

BURR-BROWN®



PCI-20007M-1

COUNTER/TIMER/PULSE GENERATOR MODULE

FEATURES

- COMPATIBLE WITH I^3 BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT
- 0.01% STABILITY
- 125ns RESOLUTION
- MULTI-FUNCTIONS
  TIME BASE GENERATOR
  PULSE GENERATION
  EVENT COUNTING, ACCUMULATING AND DECREMENTING
  FREQUENCY MEASUREMENT

DESCRIPTION

This multi-funciton module can perform a number of important time domain operations. A block diagram of the PCI-20007M-1 is shown below. Software control of the module provides an array of pulse counting and generation capabilities. Based upon an accurate 8MHz crystal-controlled oscillator, the module is useful in many precision applications. These include timebase generation, generation of finite or continuous pulse streams, event counting, accumulation and frequency measurement. The rate generator output can be linked to any other module through the I^3 bus to perform sync or other functions.

In addition to a rate generator, the module has four independent counter/timer blocks. This allows several simultaneous tasks, including multiple input and pulse generation functions.

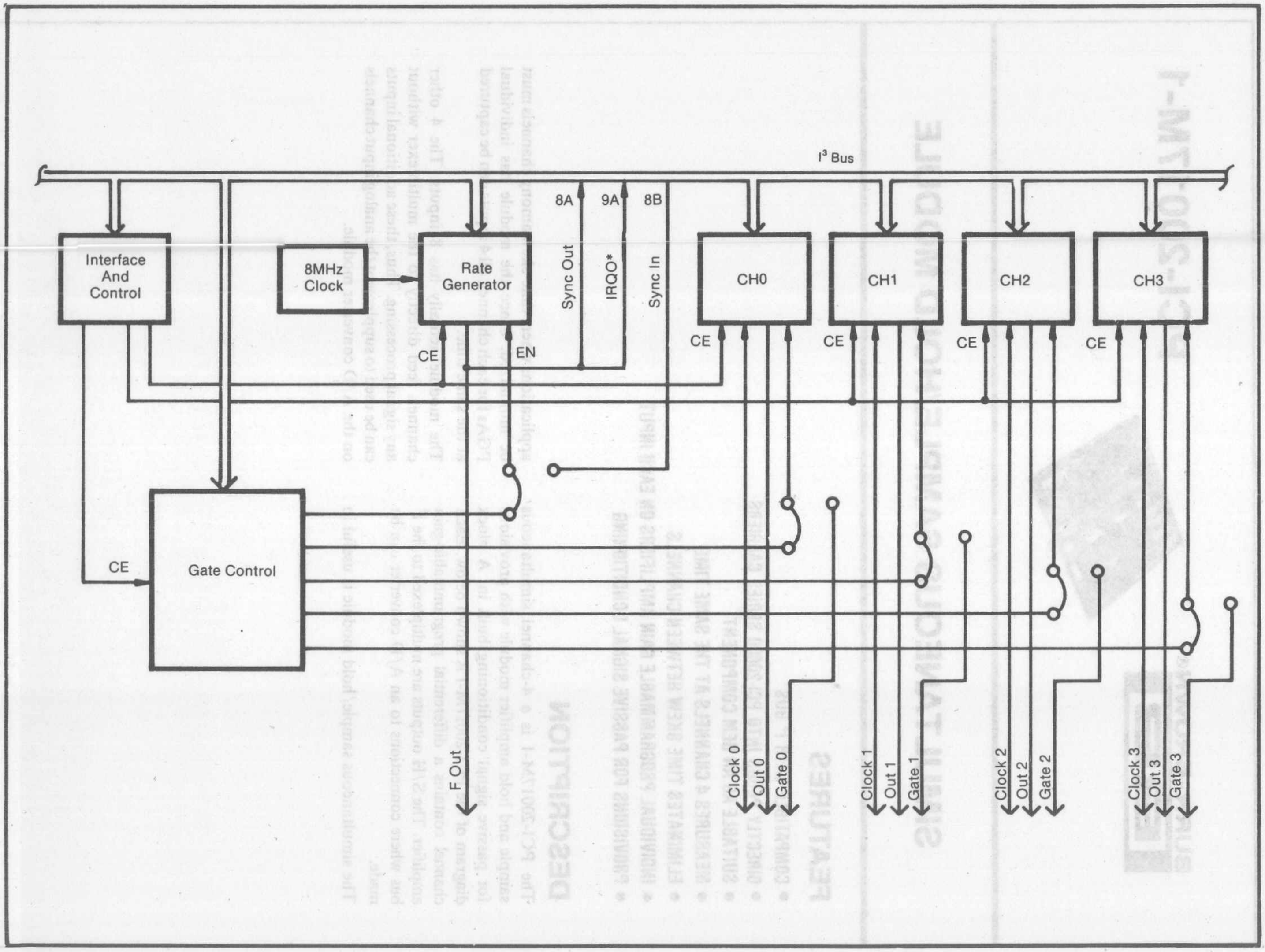
All direct input and output signals are TTL compatible. Where other levels are encountered, some applications can utilize the PCI-1100 series opto-isolators to provide logic level conversion—for example: AC line voltage switching, or remote I/O situations where ground loop connections must be broken. Field I/O connections are usually made to external termination panels, and brought to the module through ground-plane ribbon cable. The family of standard digital termination panels can be used for counter/timer applications.

SPECIFICATIONS—PCI-20007M-1

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
I/O Configuration		4 Counters and 1 Rate Generator
Functions	Counters can be preset with start value and reset when read	Count, Accumulate, Divide, Measure Frequency
Rate Generator	Pulse and square-wave outputs	
Counter Inputs		
High-Level Voltage	Minimum/Maximum	2V/5.5V
Low-Level Voltage	Minimum/Maximum	−0.5V/0.8V
I_{IN}, High-Level		200μA
I_{IN}, Low-Level		200μA
Range	16-bit counters	1 to 65,535
Sense	Pulse Input	Low-High-Low
Counter Outputs(1)		
High-Level Voltage	I_{OUT} = MAX, Minimum	2.4V
Low-Level Voltage	I_{OUT} = MAX, Maximum	0.45V
Current Source	V_{OUT} = Low	2mA
Current Sink	V_{OUT} = High	−400μA
Rate Generator(1)		
High-Level Voltage	I_{OUT} = MAX, Minimum	3.4V
Low-Level Voltage	I_{OUT} = MAX, Maximum	0.5V
Current Source	V_{OUT} = Low	8mA
Current Sink	V_{OUT} = High	−400μA
Frequency	Basic Frequency	8MHz
Range	N_1 and N_2 are 16-bit integers	8MHz/($N_1 \cdot N_2$)
Accuracy	At +25°C	±0.008%
	Over Temp Range	±0.015%
Power Requirements	+5V Supply	470mA Max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3" 9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

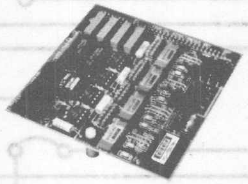
NOTE: (1) When the system is first powered up, the outputs of this module are NOT in determined states until initialized by software. That is, the outputs could be 'high' or 'low'.



PCI-20007M Module Block Diagram.

BURR-BROWN®

PCI-20017M-1



SIMULTANEOUS SAMPLE/HOLD MODULE

FEATURES

- COMPATIBLE WITH I^3 BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT
- MEASURES 4 CHANNELS AT THE SAME TIME
- ELIMINATES TIME SKEW BETWEEN CHANNELS
- INDIVIDUAL PROGRAMMABLE GAIN AMPLIFIERS ON EACH INPUT
- PROVISIONS FOR PASSIVE SIGNAL CONDITIONING

DESCRIPTION

The PCI-20017M-1 is a 4-channel simultaneous sample and hold amplifier module with provisions for passive signal conditioning built in. A block diagram of the PCI-20017M-1 is shown below. Each channel contains a differential programmable-gain amplifier. The S/H outputs are multiplexed to the I^3 bus where connections to an A/D converter can be made.

The simultaneous sample/hold module is useful in applications where time-skew among channels must be minimized. Since the module has individual PGAs for each channel, all 4 inputs can be captured at the same time.

The module actually has 8 inputs. The 4 other channels feed directly to the multiplexer without any signal processing. Thus, these additional inputs can be used to supplement the analog input channels on the A/D converter module.

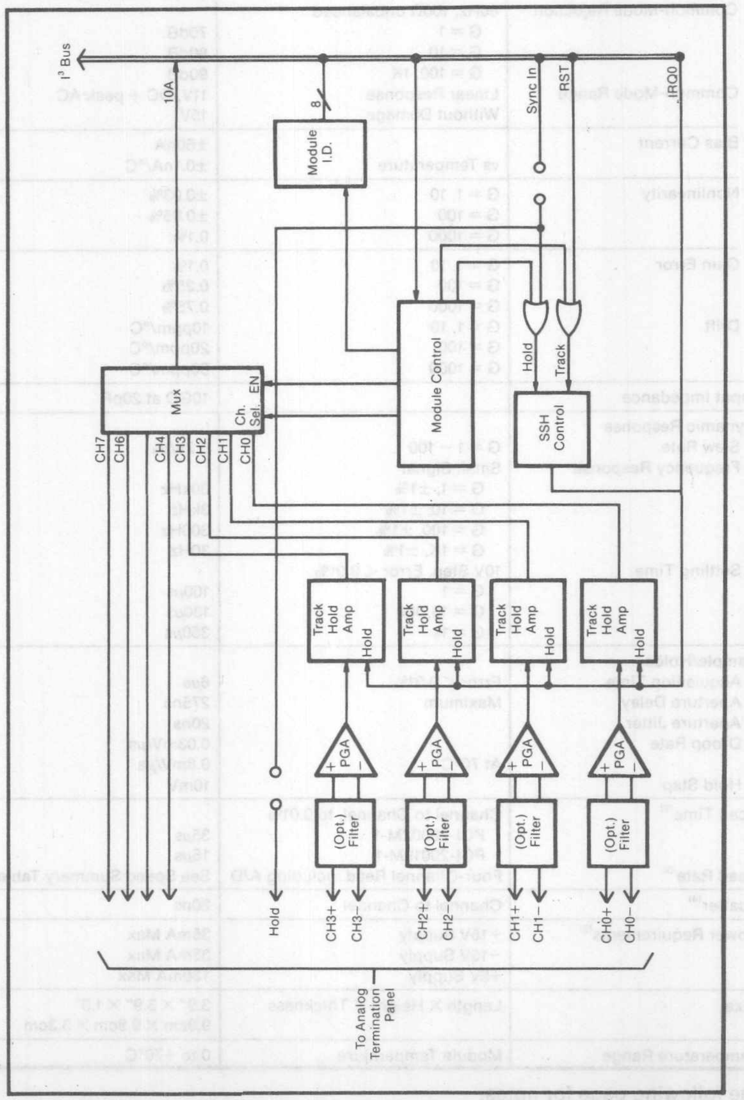
SPECIFICATIONS—PCI-20017M-1

All specifications are typical at +25°C unless noted otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
I/O Configuration		Analog Input
Number of Channels	Single-ended, Straight Thru	4
	Differential, Simultaneous	4
Input Stage[1]	Jumper-Programmable Amplifier	G = 1, 10, 100, 1K
Offset Voltage		Trimmable to zero
Offset Drift	RTI	($\pm 5\mu V \pm 10\mu V/G$)/°C
Common-Mode Rejection	60Hz, 100Ω unbalanced	
	G = 1	70dB
	G = 10	80dB
	G = 100, 1K	90dB
Common-Mode Range	Linear Response	11V, DC + peak AC
	Without Damage	15V
Bias Current		±50nA
	vs Temperature	±0.1nA/°C
Nonlinearity	G = 1, 10	±0.03%
	G = 100	±0.05%
	G = 1000	0.1%
Gain Error	G = 1, 10	0.1%
	G = 100	0.25%
	G = 1000	0.75%
Drift	G = 1, 10	10ppm/°C
	G = 100	20ppm/°C
	G = 1000	30ppm/°C
Input Impedance		10GΩ at 20pF
Dynamic Response		
Slew Rate	G = 1 − 100	$0.2V/\mu s$
Frequency Response	Small Signal	
	G = 1, ±1%	30kHz
	G = 10, ±1%	3kHz
	G = 100, ±1%	300Hz
	G = 1K, ±1%	30Hz
Settling Time	10V Step, Error < 0.01%	
	G = 1	$100\mu s$
	G = 10, 100	$130\mu s$
	G = 1k	$350\mu s$
Sample/Holds		
Acquisition Time	Error < 0.01%	$6\mu s$
Aperture Delay	Maximum	275ns
Aperture Jitter		20ns
Droop Rate		$0.03mV/\mu s$
	At 70°C	$0.8mV/\mu s$
Hold Step		10mV
Scan Time[2]	Channel to Channel, to 0.01%	
	PCI-20002M-1	$35\mu s$
	PCI-20019M-1	$15\mu s$
Read Rate[3]	Four-Channel Read including A/D	See Speed Summary Table
Scatter[4]	Channel to Channel	20ns
Power Requirements[5]	+15V Supply	35mA Max
	−15V Supply	35mA Max
	+5V Supply	130mA Max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3"
		9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

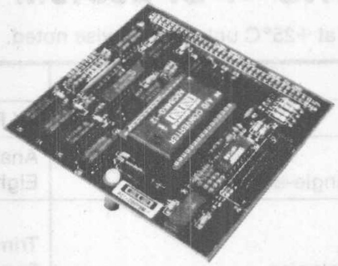
See following page for notes.

NOTES: (1) This applies to the differential channels only. (2) 'Scan time' is defined as the time required to select one of the four S/H channels and to read it with a given A/D converter. (3) 'Read Rate' is defined as the rate at which S/H channels can be read using the PCI-20046S/47S High Speed Read, expressed on a per-channel basis. It is assumed that all four channels hold desired data. (4) 'Scatter' is defined as the maximum **difference** in time required to capture all S/H channels. It is a measure of the system's 'simultaneity'. This is the key specification of a simultaneous S/H system. (5) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 550mA, maximum. This takes into account the efficiency of the DC/DC converter.



PCI-20017M-1 Four-Channel Simultaneous Sample/Hold Module Block Diagram.

BURR-BROWN®



PCI-20019M-1

HIGH SPEED DATA ACQUISITION MODULE

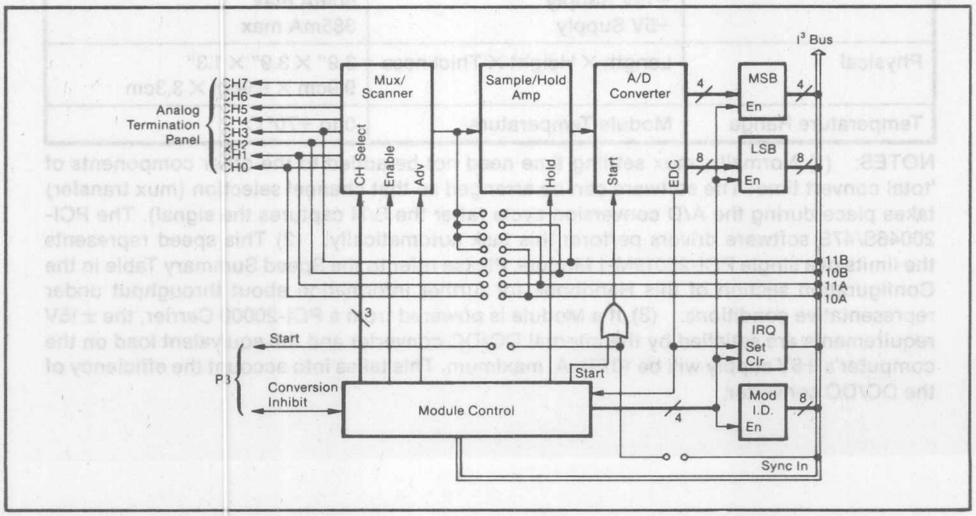
FEATURES

- **Directly plugs into PCI-20000 Series Carriers**
- **89kHz throughput rate**
- **Eight-channel input**
- **Hardware and software trigger capability**
- **Automatic channel advance**
- **Suitable as an OEM component**

DESCRIPTION

The PCI-20019M-1 is a high speed, 12-bit data acquisition module. Eight single-ended input channels are provided. This module is intended for high-level signals and does not contain an input amplifier which could reduce its speed. The combination of a high speed sample/hold and A/D converter provides for input sampling at about 89,000 channels/second. The full-scale input range can be jumper-selected for 0 to +5V, 0 to +10V, ±2.5V, ±5V or ±10V. Internal hardware can configure the module to automatically increment channels after each 'start convert'. This feature greatly reduces the computer's software burden and results in increased speed. Conversions may be started from either an internal or external signal, upon reading the previous conversion, or by software command.

Additional input channels can be obtained by using the optional PCI-20031M-1 Expansion Module. Each expander adds 32 channels. The PCI-20019M-1 is also compatible with the PCI-20017M-1 Simultaneous Sample/Hold Module and with the PCI-20020M-1 Trigger/Alarm Module.



PCI-20019M Module Block diagram.

SPECIFICATIONS—PCI-20019M-1

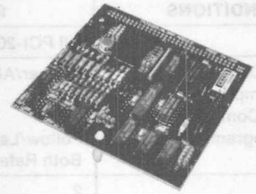
All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
I/O Configuration Number of Channels	 Single-ended	Analog input Eight
Input Offset Voltage Offset Drift Impedance Voltage Range Bias Current Noise	 Unipolar Bipolar Linear Without Damage	 Trimmable to zero 3ppm/°C 15ppm/°C 1MΩ at 35pf ±10V 20V above supplies 100nA ±1LSB
A/D Converter Resolution Code Linearity Error Drift Gain Accuracy Drift Ranges	 Unipolar Bipolar	 12 bits Complementary Binary Complementary Offset Binary ±0.5LSB ±3ppm/°C Trimmable to zero ±30ppm/°C ±2.5V, ±5V, ±10V, 0-5V, 0-10V
Dynamic Response Mux Settling Time Conversion Time Start Jitter Acquisition Time Total Convert Time Throughput Rate	 Within 0.01%, Maximum A/D, Maximum Start Convert Uncertainty S/H, Maximum	 3.5μs[1] 10μs 200ns 1.5μs 11.25μs 89k channels/second[2]
Power Requirements[3]	+15V Supply −15V Supply +5V Supply	65mA max 50mA max 385mA max
Physical	Length × Height × Thickness	3.9" × 3.9" × 1.3" 9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

NOTES: (1) Normally, mux settling time need not be added to the other components of 'total convert time'. The software can be arranged so that channel selection (mux transfer) takes place during the A/D conversion cycle (after the S/H captures the signal). The PCI-20046S/47S software drivers perform this task automatically. (2) This speed represents the limits of a single PCI-20019M-1 Module. Please refer to the Speed Summary Table in the Configuration section of this Handbook for further information about throughput under representative conditions. (3) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 1075mA, maximum. This takes into account the efficiency of the DC/DC converter.

BURR-BROWN®

PCI-20020M-1



TRIGGER/ALARM MODULE

FEATURES

- COMPATIBLE WITH I³ BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT
- SOFTWARE PROGRAMMABLE LIMITS
- DUAL CHANNEL
- HIGH, LOW OR WINDOW COMPARISONS
- 3.5μsec RESPONSE TIME (MAX)

DESCRIPTION

The PCI-20020M-1 Trigger/Alarm module can monitor 1 or 2 analog channels, and will generate a digital output when pre-programmed conditions are satisfied. A block diagram is shown below. Thresholds in the range of ±10V can be programmed with 8-bit resolution. A trigger can be initiated on one of the following conditions:

Input BELOW limit,
Input ABOVE limit,
Input BETWEEN limits or
Input OUTSIDE limits.

A pair of D/A converters and comparators are provided to perform the above functions. In the window modes (inputs Between or Outside), both comparators are connected to a single input. In all modes of operation the comparator outputs are combined with logic to produce a single output. The module can be programmed to trigger on true or false conditions. To minimize any oscillations or erroneous triggering, the comparators are designed with approximately 25mV or 1/2LSB of hysteresis. Both of the individual DAC and comparator outputs are available on the termination panel.

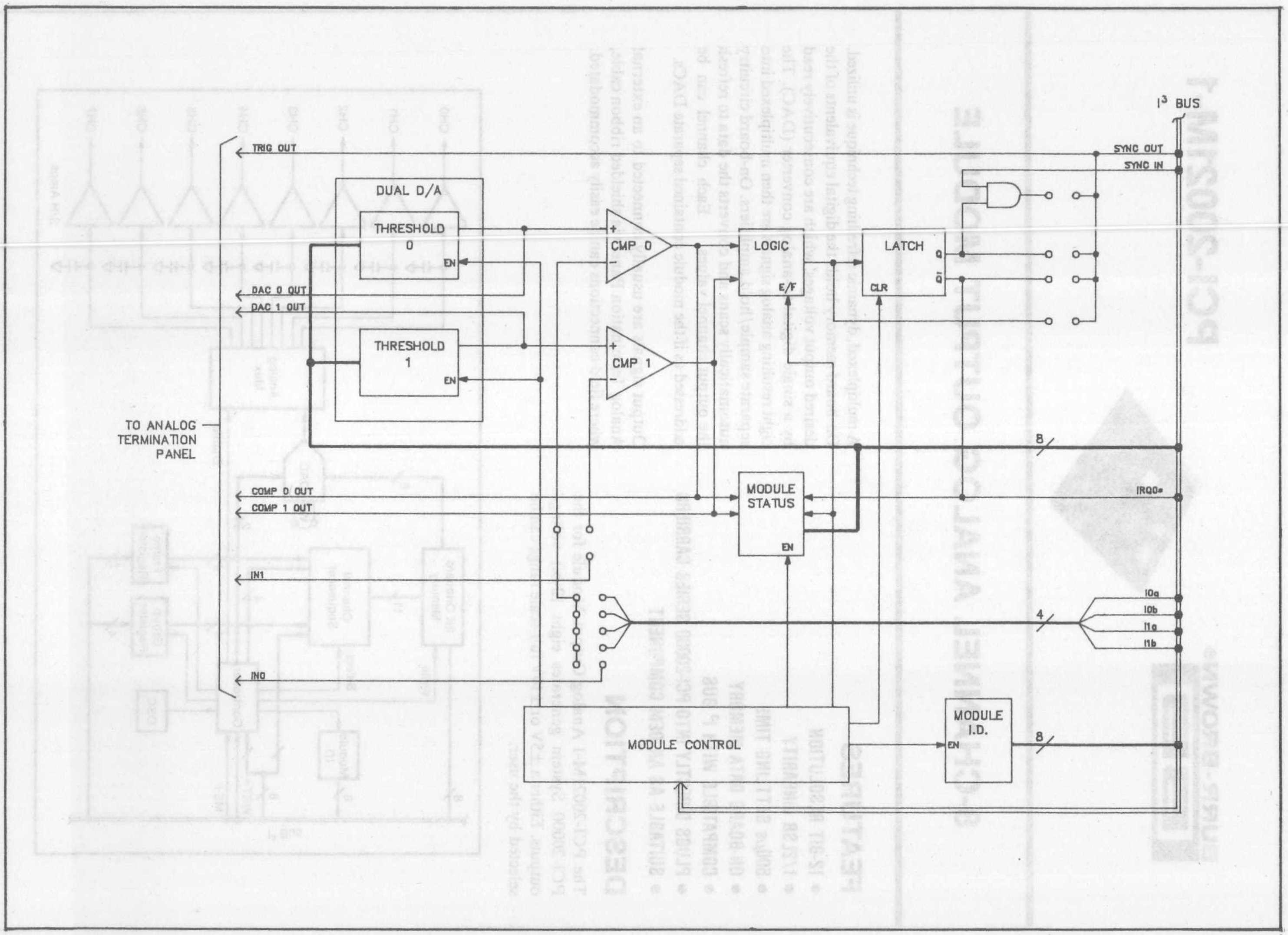
Jumper options select the mode of operation, gating of the digital output and whether or not the output is latched. Once latched, the alarm indication will remain until cleared by software.

SPECIFICATIONS—PCI-20020M-1

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
I/O Configuration	Analog or Digital Inputs	Trigger/Alarm
Number of Channels	Level Compare	2
	Window Compare	1
Sync Output(1)	Jumper Programmable	Follow/Latched, Gated Sync In
Analog Outputs(1)		Both References and Comparators
Comparators		2
Input Range	Linear	±10V
	Without Damage	±15V
Bias Current		300nA
Offset Voltage	Maximum	±7.5mV
Hysteresis	±10%	25mV
References	D/A Converters	2
Resolution		8 Bits
Step Size	Minimum Increment	78.1mV
Code		Offset Binary
Input Range	Maximum	+9.92V, −10V
Linearity		±1/2LSB
Response Time	Input to Sync Output, Max	3.5μs
Power Requirements(2)	+15V Supply	35mA Max
	−15V Supply	25mA Max
	+5V Supply	265mA Max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3" 9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

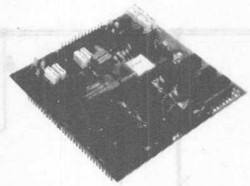
NOTE: (1) When the system is first powered up, the outputs of this module are NOT in determined states until initialized by software. That is, the analog outputs could be any value between ±10V and the digital outputs could be either 'high' or 'low'. (2) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 625mA, maximum. This takes into account the efficiency of the DC/DC converter.



PCI-20002M Module Block Diagram.

BURR-BROWN®

PCI-20021M-1



8-CHANNEL ANALOG OUTPUT MODULE

FEATURES

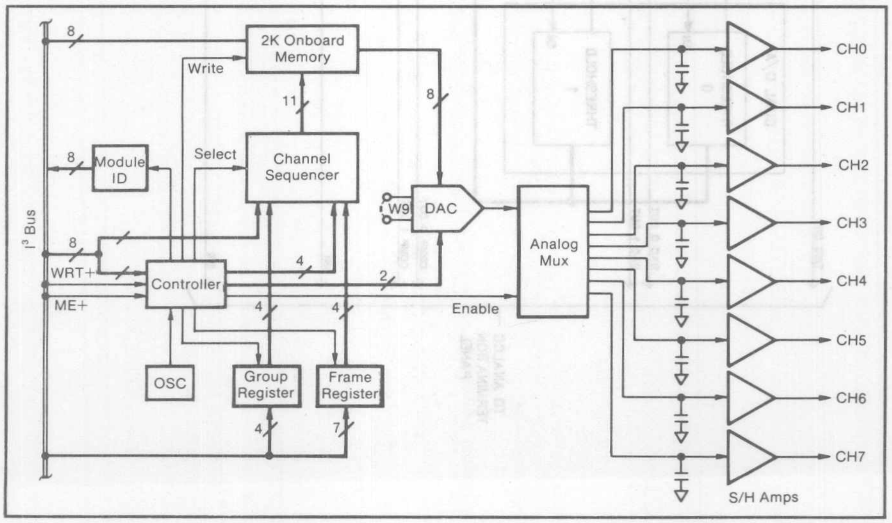
- 12-BIT RESOLUTION
- 1/2LSB LINEARITY
- 500μs SETTLING TIME
- ON-BOARD DATA MEMORY
- COMPATIBLE WITH I³ BUS
- PLUGS DIRECTLY INTO PCI-20000 SERIES CARRIERS
- SUITABLE AS AN OEM COMPONENT

DESCRIPTION

The PCI-20021M-1 Analog Output Module for the PCI-20000 System generates eight 12-bit voltage outputs. Either a ±5V or ±10V full-scale range can be selected by the user.

A multiplexed, dynamic refreshing technique is utilized. On-board memory holds the digital equivalents of the desired output voltages, which are consecutively read by a single digital-to-analog converter (DAC). The eight resulting analog signals are then multiplexed into separate sample/hold amplifiers. On-board circuitry automatically scans and converts the data to refresh the output channel values. Each channel can be addressed as if the module contained separate DACs.

Output signals are usually connected to an external Analog Termination Panel via shielded ribbon cable, where field connections can be easily accommodated.



SPECIFICATIONS—PCI-20021M-1

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		All PCI-20000 Carriers
Configuration:		Analog output
Range[1]	Jumper selectable	±5V, ±10V
Number of Channels		Eight
Resolution		12 bits
Code		Offset binary
RAM Data Buffer	Eight Channels Each Frame	128 frames
Linearity		±1/2LSB
Gain Accuracy		±1/2LSB
Drift		±30ppm/°C
Offset		±2mV
Drift		±15ppm/°C
Output Stage:		
Current		±1mA
Impedance	At 2kHz	1Ω
Settling Time		500μs
Refresh Time	8-Channel Cycle	128μs
Conversion Rate		2kHz
Noise	DC to 10kHz, Maximum	±1LSB
Feedthrough	Channel to Channel	±1LSB
Power Requirements[2]	+15V Supply	43mA max
	−15V Supply	50mA max
	+5V Supply	569mA max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3"
		9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

NOTES: (1) When the system is first powered up, the outputs of this module are NOT in determined states until initialized by software. That is, the analog outputs could be any value consistent with the hardware jumpers installed. (2) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 1127mA, maximum. This takes into account the efficiency of the DC/DC converter.

BURR-BROWN®

PCI-20023M-1

ADVANCE INFORMATION
Subject to Change

HIGH SPEED DATA ACQUISITION MODULE

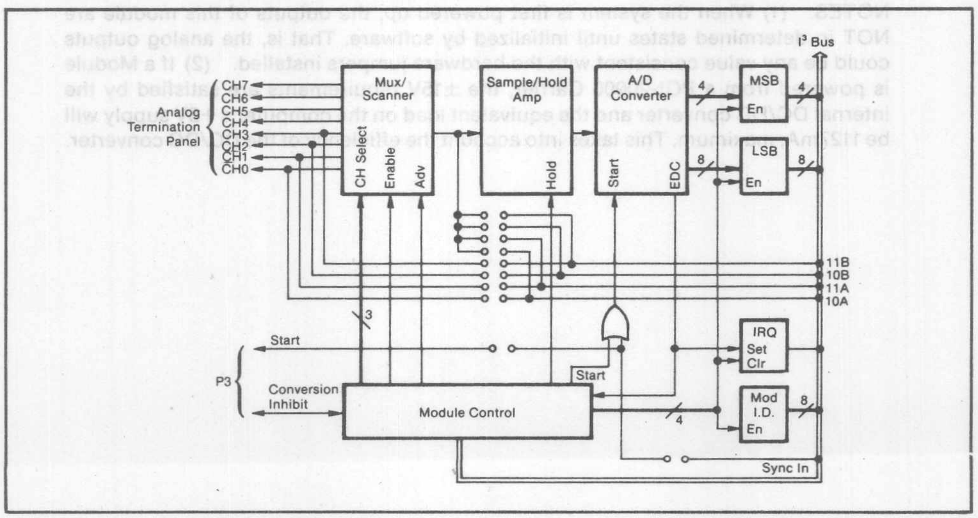
FEATURES

- **Directly plugs into PCI-20000 Series Carriers**
- **180kHz throughput rate in a PC/XT using DMA**
- **Eight-channel input**
- **Hardware and software trigger capability**
- **Automatic channel advance**
- **Suitable as an OEM component**

DESCRIPTION

The PCI-20023M-1 is a high speed, 12-bit data acquisition module. Eight single-ended input channels are provided. This module is intended for high-level signals and does not contain an input amplifier which could reduce its speed. The combination of a high speed sample/hold and A/D converter provides for input sampling at about 180,000 channels/second. The full-scale input range can be jumper-selected for 0 to +10V, ±5V, or ±10V. Internal hardware can configure the module to automatically increment channels after each 'start convert'. This feature greatly reduces the computer's software burden and results in increased speed. Conversions may be started from either an internal or external signal, upon reading the previous conversion, or by software command.

Additional input channels can be obtained by using the optional PCI-20031M-1 Expansion Module. Each expander adds 32 channels. The PCI-20023M-1 is also compatible with the PCI-20017M-1 Simultaneous Sample/Hold Module and with the PCI-20020M-1 Trigger/Alarm Module.



SPECIFICATIONS—PCI-20023M-1

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
COMPATIBILITY		All PCI-20000 carriers
I/O CONFIGURATION		Analog input
Number of Channels	Single-ended	Eight
INPUT		
Offset Voltage		Trimmable to zero
Offset Drift	Unipolar	3ppm/°C
	Bipolar	15ppm/°C
Impedance		1MΩ
Voltage Range	Linear	±10V
	Without damage	20V above supplies
Bias Current		100nA typ, 300nA max
Noise		±1LSB on 5V range
A/D CONVERTER		
Resolution		12 bits
Code	Unipolar	Binary
	Bipolar	Offset binary
Linearity Error		±0.5LSB
Drift		±3ppm/°C
Gain Accuracy		Trimmable to zero ±1/2LSB
Drift		±30ppm/°C
Ranges		±5V, ±10V, 0-10V
DYNAMIC RESPONSE		
Mux Settling Time	Within 0.01%, max	3.5μs[1]
Conversion Time	A/D, max	4μs
Aperture Jitter		0.3μs
Acquisition Time	S/H, max	1.5μs
Total Convert Time		5.5μs
Throughput Rate	In an IBM PC/XT using DMA	180k channels/second
POWER REQUIREMENTS[2]	+15V Supply	42mA max
	−15V Supply	52mA max
	+5V Supply	685mA max
PHYSICAL	Length × height × thickness	3.9" × 3.9" × 1.3" (9.9cm × 9.9cm × 3.3cm)
TEMPERATURE RANGE	Module temperature	0°C to +70°C

NOTES: (1) Normally, mux settling time need not be added to the other components of 'total convert time'. The software can be arranged so that channel selection (mux transfer) takes place during the A/D conversion cycle (after the S/H captures the signal). The PCI-20046S/47S software drivers perform this task automatically. (2) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 1250mA, maximum. This takes into account the efficiency of the DC/DC converter.

BURR-BROWN®

PCI-20031M-1

ANALOG EXPANDER/SEQUENCER MODULE

FEATURES

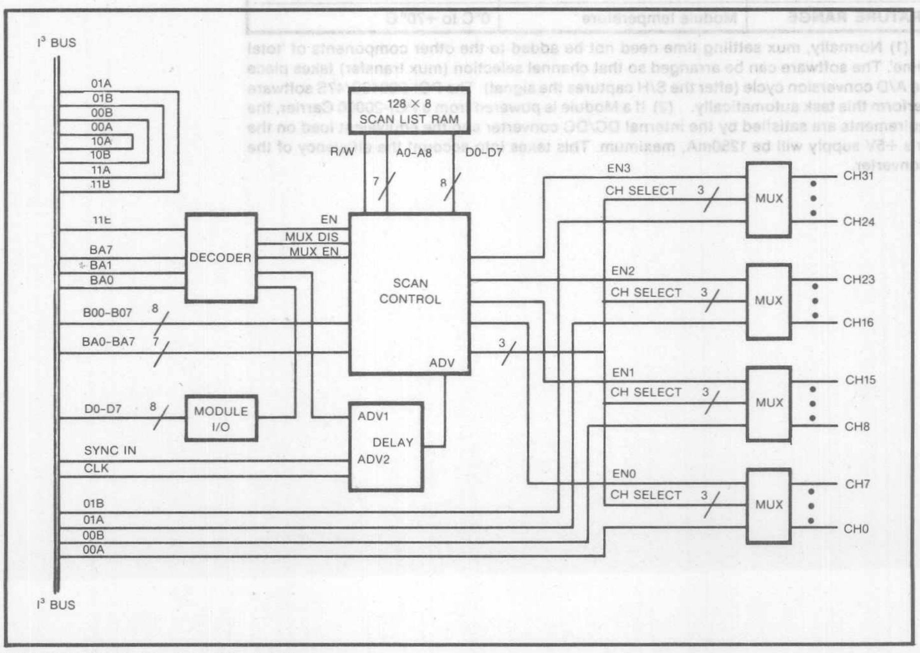
- ADDS 32 CHANNELS TO EXISTING ANALOG INPUT COUNT
- HIGH SPEED—UP TO 89kHz SCAN RATE
- SCAN LIST HELD IN ON-BOARD MEMORY
- SCAN LIST CAN CONTAIN UP TO 128 ELEMENTS
- INPUTS PROTECTED TO ±20V
- SUITABLE AS AN OEM COMPONENT
- COMPATIBLE WITH I^3 BUS
- DIRECTLY PLUGS INTO PCI-20000 SERIES CARRIERS

APPLICATIONS

- DIGITAL OSCILLOSCOPE
- HIGH SPEED DATA LOGGER
- FAST DIGITAL VOLTMETER
- WAVEFORM RECORDING
- VIBRATION ANALYSIS

DESCRIPTION

The Analog Expander/Sequencer Module was designed to complement the PCI-20019M-1 High Speed Analog Data Acquisition Module in the PCI-20000 System by providing 32 additional multiplexed input channels. The throughput speed is limited by the Data Acquisition Module and the host computer, and is greater than 50kHz in the IBM PC and up to 89kHz in the IBM AT. The PCI-20031M-1 can also be used with the PCI-20002M-1 to add 32 single-ended or 16 differential input channels. In this respect, the PCI-20031M-1 is similar to the PCI-20005M-1. When used with the PCI-20019M-1, high speed performance is enhanced by



PCI-20031M-1 Module Block Diagram.

the use of on-board memory to store the desired scan list and hardware to automatically advance through the scan list. This can be accomplished via a software, internal hardware or external hardware signal (for example, each time an A/D conversion is initiated). This allows scanning to be accomplished without intervention from the host computer. The scan list can store up to 128 elements. The channel sequence specified in the scan list can be in any order desired, and can include duplicate channels. Duplicate channels are extremely useful in cases where a given channel must be sampled at a higher frequency than others.

SPECIFICATIONS—PCI-20031M-1

All specifications are typical at +25°C unless otherwise noted.

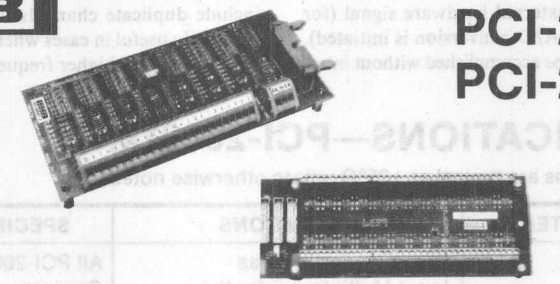
PARAMETER	CONDITIONS	SPECIFICATION
Compatibility	Primarily designed as Input Multiplexers for the PCI-20002M / 19M A/D Modules	All PCI-20000 Carriers
I/O Configuration		Analog Expander
Number of Channels	Single-ended	32
	Differential (with PCI-20002M)	16
Analog Signal Range	Linear operation	±10V
	Without Damage	20V above supply
Input Capacitance[1]	Channel 'On': Single-ended	100pF
	Differential	50pF
	Channel 'Off'	5pF
'On' Resistance	Maximum	1.8kΩ
'Off' Isolation	Frequency = 1kHz, R_S = 1kΩ	85dB
Input Leakage	'On' Channel: at 25°C	0.1nA
	at 70°C	2.5nA
	'Off' Channel: at 25°C	.03nA
	at 70°C	0.7nA
Channel List	On-Board RAM, advanced with Sync input or S/W command	
Length		128 Channels
Power Requirements[2]	+15V Supply	8mA max
	−15V Supply	4mA max
	+5V Supply	345mA max
Size	Length × Height × Thickness	3.9" × 3.9" × 1.3"
		9.9cm × 9.9cm × 3.3cm
Temperature Range	Module Temperature	0 to +70°C

NOTES: (1) For a single 31M module without regard to 'loads' connected through the I^3 Bus. However, in the single-ended mode, it is assumed that all four mux outputs are connected together. In the differential mode, two mux outputs are connected together.

(2) If a Module is powered from a PCI-20000 Carrier, the ±15V requirements are satisfied by the internal DC/DC converter and the equivalent load on the computer's +5V supply will be 417mA, maximum. This takes into account the efficiency of the DC/DC converter.

BURR-BROWN®

PCI-20010T-1
PCI-20010T-2
PCI-20057T-1



ANALOG TERMINATION PANELS

FEATURES

- **Clamp type screw terminals for field wiring connections**
- **Extra terminals for ground and two-wire transmitter connections**
- **Extensive, passive, signal conditioning capabilities**
- **Thermocouple, cold-junction monitor (except PCI-20010T-1)**
- **Can be used for either input or output functions**
- **Compatible with all analog PCI and selected VME products**
- **Suitable as an OEM component**

DESCRIPTION

The PCI-20010T Termination Panels can accommodate up to 16 channels of single-ended analog inputs or outputs, with signal conditioning available on each channel. Differential inputs may be connected by using the single-ended channels in pairs, thus allowing up to eight differential inputs per panel. One ribbon cable connector provides the interface for all 8/16 channels to the DA&C system.

The PCI-20057T-1 Termination Panel can accommodate up to 48 channels of single-ended analog inputs or outputs, with signal conditioning available on each channel. Differential inputs may be connected by using the single-ended channels in pairs, thus allowing up to 24 differential inputs per panel. Each group of 16 single-ended channels is interfaced to the DA&C system with a separate ribbon cable connector (a total of three connectors).

The Termination Panels are divided into identical circuits (eight for the PCI-20010Ts and 24 for the PCI-20057T-1); each circuit is associated with a set of screw-terminal blocks at the edge of the panel. Figure 1 shows a block diagram of the PCI-20010T-2 panel, which is representative of all three panels.

Provisions for the user to install passive signal conditioning networks is included. The printed-circuit layout accommodates current-to-voltage conversion, voltage dividers, filters, surge suppression, and open thermocouple detection, etc. Figure 2 shows the physical layout of the PCI-20057T-1. The PCI-20010T panels are similar in concept, with fewer terminals and connectors.

The PCI-20010T-2 is intended primarily for thermocouple applications. However, the function of all channels can be intermixed. The panel is factory configured for seven differential connections. Circuitry to monitor the ambient temperature of the screw terminals, for cold-junction compensation purposes, is included. This sensor is wired to channel four. Bias current return resistors for all seven thermocouple channels are installed.

The PCI-20057T-1 also has a cold-junction compensation monitor and is thus very useful for thermocouple applications. Up to 23 thermocouples can be supported on one PCI-20057T-1 panel. Bias current return resistors are installed on all channels.

Field connections are made to the panels via the screw-terminal blocks. There are three screw terminals per pair of I/O channels. The center terminal on the PCI-20057T-1 is ground. On the PCI-20010Ts, the terminal between each I/O pair is jumper pro-

grammable as either a ground or +V connection. The ground terminals greatly facilitate single-ended connections, while the +V option eases connections in two-wire transmitter applications (a separate block is provided on the PCI-20010Ts for the connection of an external-excitation or current-loop power supply). Connections to other parts of the DA&C system are made using the 26-pin ribbon type connectors provided. All panels are compatible with PCI-20012A shielded cables, the PCI-20032A-1 multi-module cable and PCI-20029A rack mount enclosures.

SPECIFICATIONS

PCI-20010T-1, PCI-20010T-2, PCI-20057T-1

All specifications are typical at +25°C, unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Function	Analog, Input/Output, Passive Signal Conditioning Termination Panel	
I/O Configuration	PCI-20010T-1, single-ended Differential PCI-20010T-2, Differential Thermocouple PCI-20057T-1, Single-ended Differential Thermocouple	16 channels 8 channels 8 channels 7 channels(1) 48 channels 24 channels 23 channels(1)
Terminal/Connections: Field Wires System Connection	 Terminal Type Wire Size Range Number of Terminals Configuration: PCI-20010T PCI-20057T-1 26-pin Mating Connector	 Screw clamp 22-12 AWG 27 Channel, X, +V, Ground(2) Channel, ground Ansley #609-2630
Signal Conditioning(3)	I/V Conversion, Voltage Division, Surge Protection, Filters(4), Bridge Completion, Cold-Junction Compensation (except PCI-20010T-1)	
Cold-Junction Compensation	PCI-20010T-2 and PCI-20057T-1	1mV/°K, ±1°K
Power Available	System Power, fused at 0.25A	Via ribbon cable(5)
Size	Length × Width × Height PCI-20010T PCI-20057T-1	 8″ × 3.5″ × 1.6″ 20.3cm × 8.9cm × 4.1cm 9″ × 3.5″ × 1.6″ 22.9cm × 8.9cm × 4.1cm
Temperature Range	Panel Temperature	0 to +70°C

: (1) When used with thermocouples, differential connections are assumed. On the PCI-20010T-2, the output of the cold-junction monitor is connected to channel four, leaving seven channels available for use. On the PCI-20057T-1, the output of the cold-junction monitor is connected to channel zero, leaving 23 channels available for use. (2) On the PCI-20010T panels only, the X terminals are jumper programmable as either Ground or +V. +V is an externally connected power supply intended for applications such as 4-20mA loops, bridge excitation, etc. (3) The Termination Panels' printed circuit pattern is designed to accept user-defined passive signal conditioning networks. Other than the 100kΩ bias-current-return resistors on the PCI-20010T-2 and PCI-20057T-1, these components are not provided. See the Signal Conditioning section of this Handbook for detailed information on recommended components. (4) The PCI-20010Ts have room for one or two poles of passive filtering on each channel. The PCI-20057T-1 has room for one pole of filtering per channel. (5) When used with PCI-20000 Modules, the System connection cable brings +5V from the computer supply to the termination panel. This is used to power the cold-junction monitor (PCI-20010T-2 and PCI-20057T-1), but is also available for other applications.

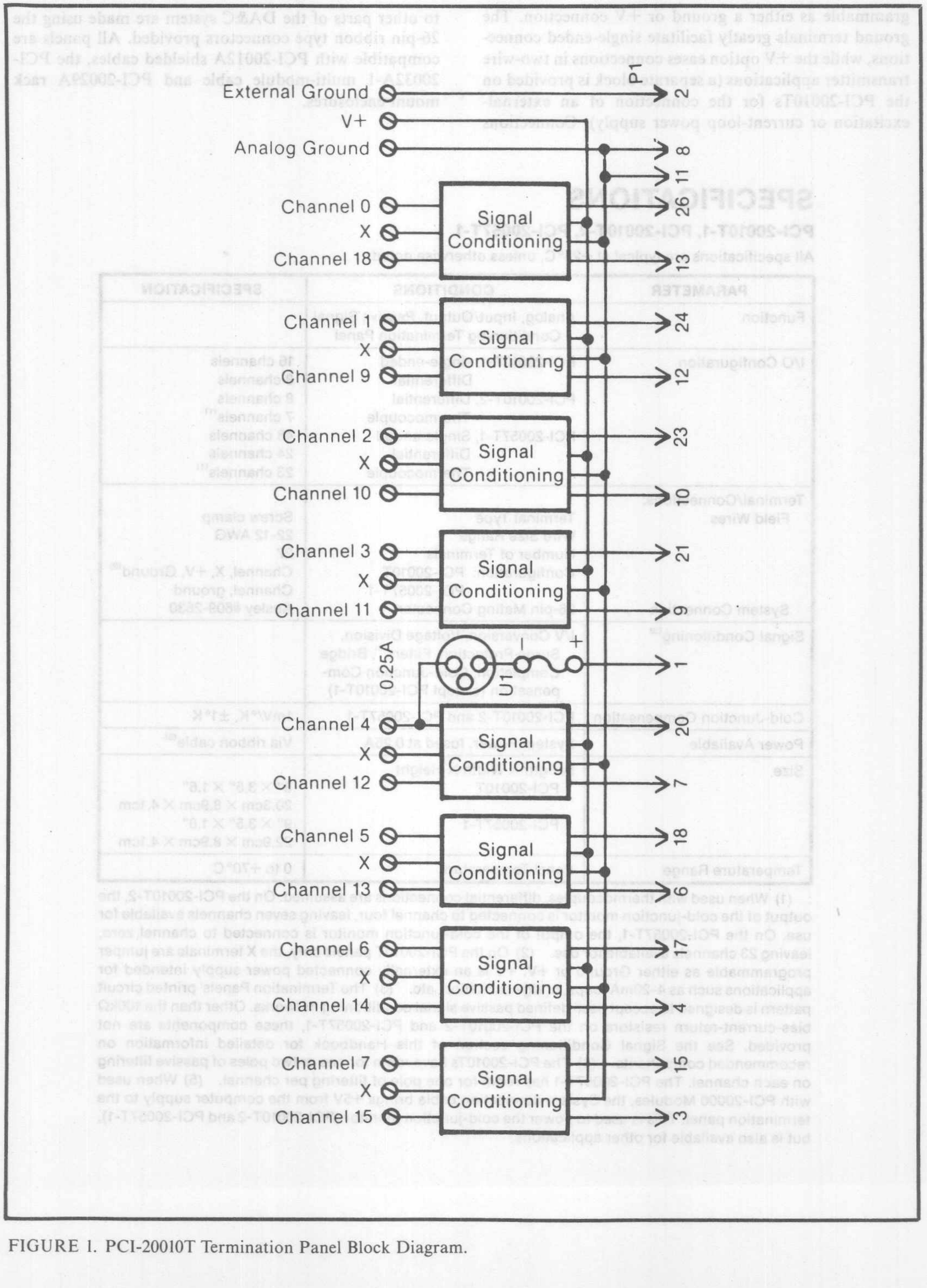


FIGURE 1. PCI-20010T Termination Panel Block Diagram.

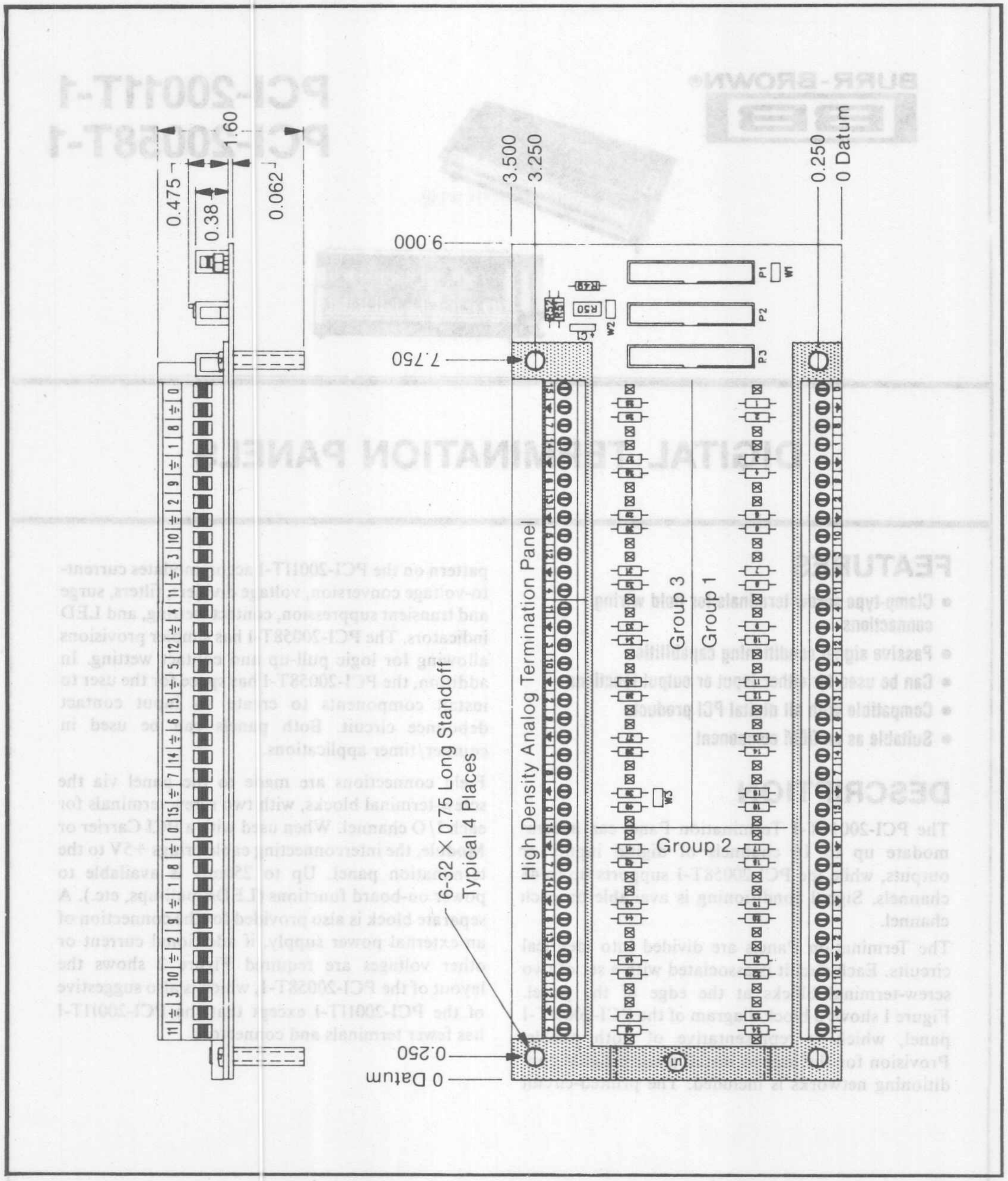
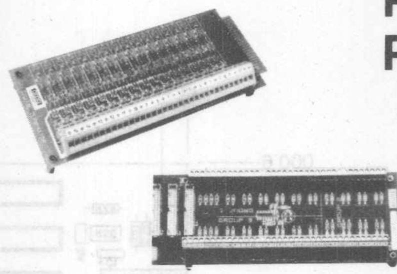


FIGURE 2. PCI-20057T Termination Panel Layout and Dimensions (inches).

BURR-BROWN®

PCI-20011T-1
PCI-20058T-1



DIGITAL TERMINATION PANELS

FEATURES

- **Clamp-type screw terminals for field wiring connections**
- **Passive signal conditioning capabilities**
- **Can be used for either input or output functions**
- **Compatible with all digital PCI products**
- **Suitable as an OEM component**

DESCRIPTION

The PCI-20011T-1 Termination Panel can accommodate up to 16 channels of digital inputs or outputs, while the PCI-20058T-1 supports up to 48 channels. Signal conditioning is available on each channel.

The Termination Panels are divided into identical circuits. Each circuit is associated with a set of two screw-terminal blocks at the edge of the panel. Figure 1 shows a block diagram of the PCI-20011T-1 panel, which is representative of both panels. Provision for the user to install passive signal conditioning networks is included. The printed-circuit pattern on the PCI-20011T-1 accommodates current-to-voltage conversion, voltage dividers, filters, surge and transient suppression, contact wetting, and LED indicators. The PCI-20058T-1 has simpler provisions allowing for logic pull-up and contact wetting. In addition, the PCI-20058T-1 has space for the user to install components to create an input contact debounce circuit. Both panels can be used in counter/timer applications.

Field connections are made to the panel via the screw-terminal blocks, with two screw terminals for each I/O channel. When used with a PCI Carrier or Module, the interconnecting cable brings +5V to the termination panel. Up to 250mA is available to power on-board functions (LEDs, pull-ups, etc.). A separate block is also provided for the connection of an external power supply, if additional current or other voltages are required Figure 2 shows the layout of the PCI-20058T-1, which is also suggestive of the PCI-20011T-1 except that the PCI-20011T-1 has fewer terminals and connectors.

SPECIFICATIONS—PCI-20011T-1, PCI-20058T-1

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Function	Digital, Input/Output, 'Passive' Signal Conditioning Termination Panel	
I/O Configuration	PCI-20011T-1 PCI-20058T-1	16 channels 48 channels
Signal Conditioning		User installed[1]
Terminal Connections: Field Wires	Terminal Type Wire Size Range Number of Terminals: PCI-20011T-1 PCI-20058T-1 Configuration	Screw clamp 22–12 AWG 34 73 Ground, channel, channel
System Connection	Mating Connector: PCI-20011T-1, one required PCI-20058T-1, three required	 Ansley #609-5015M Ansley #609-3430
Power Available	System Power, Fused at 0.25A	Via ribbon cable[2]
Size	Lenght × Width × Height: PCI-20011T-1 PCI-20058T-1	 8" × 3.5" × 1.6" 20.3cm × 8.9cm × 4.1cm 9" × 3.5" × 1.6" 22.9cm × 8.9cm × 4.1cm
Temperature Range	Panel Temperature	0 to 70°C

NOTES: (1) The Termination Panels' printed-circuit pattern is designed to accept user-defined passive signal-conditioning networks. These components, however, are not provided. See the Signal Conditioning section of this Handbook for detailed information on recommended components. (2) When used with PCI-20000 Carriers or Modules, the system connection cable brings +5V from the computer supply to the termination panel. This can be used to power on-board LEDs (PCI-20011T-1 only) for pull-ups and contact wetting, etc.

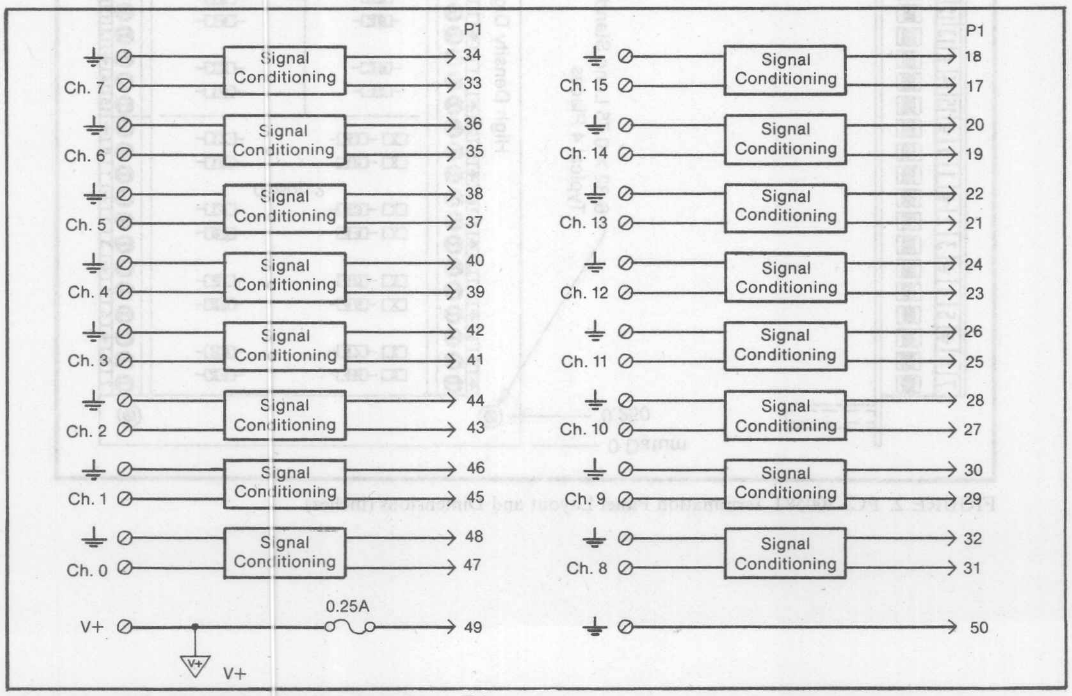


FIGURE 1. PCI-20011T Termination Panel Block Diagram.

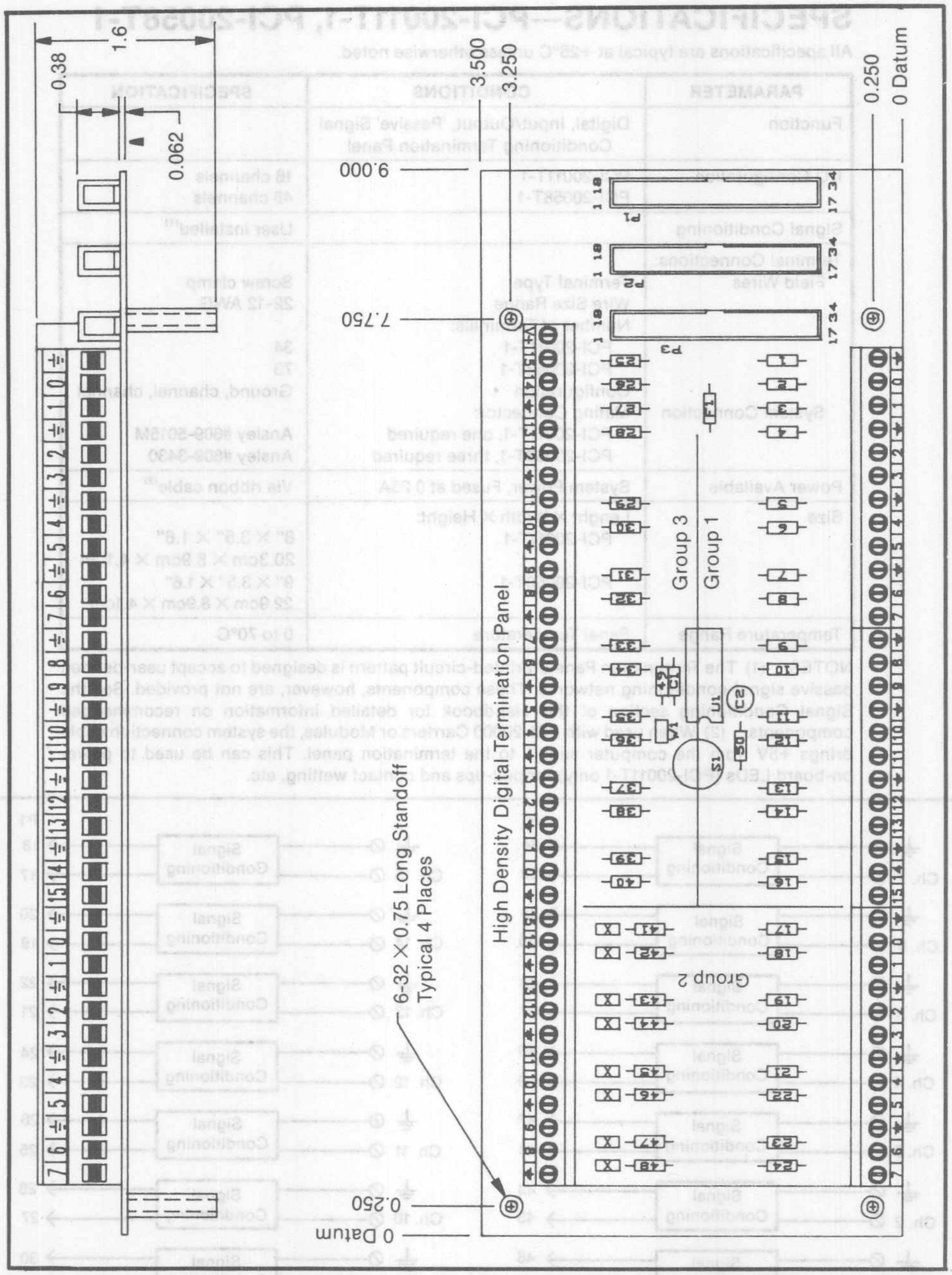
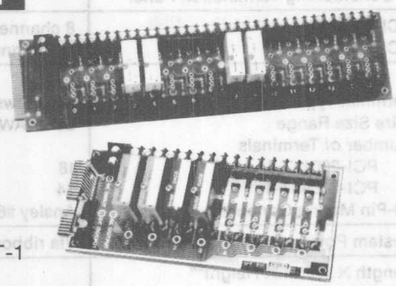


FIGURE 2. PCI-20058T Termination Panel Layout and Dimensions (inches).

BURR-BROWN®

PCI-20018T-1
PCI-20048T-1

PCI-20048T-1



PCI-20018T-1

OPTICALLY ISOLATED DIGITAL TERMINATION PANELS

FEATURES

- Screw terminals provide for easy field wiring connections
- Can be used for either input or output functions
- Compatible with PCI-1100 Series Opto-Isolators
- Interconnects with all digital PCI products
- LEDs indicate channel status
- Suitable as an OEM component

DESCRIPTION

The PCI-20018T-1 and PCI-20048T-1 are Signal Termination Panels that accommodate the PCI-1100 Series, Optically Isolated I/O Modules. Various combinations of the different Opto Modules can be intermixed on one panel. However, the design of the PCI-20000 digital I/O ports define that each group of eight channels must be either inputs or outputs. The PCI-20018T-1 supports up to eight isolated channels, while the PCI-20048T-1 supports up to 16 isolated channels. These panels can also be used in Counter/Timer applications.

The Termination Panels are divided into either eight or 16 identical circuits, each circuit being associated with a set of two screw-terminal blocks at the edge of the panel. Figure 2 shows a schematic diagram of the PCI-20048T-1 panel.

Field connections are made to the panel via the screw-terminal blocks. When used with a PCI Carrier or Module, the interconnecting cable brings +5V to the termination panel. Up to 250mA is available to power on-board functions, including the LEDs. A separate block is provided for the connection of an external power supply if this is desired. A 50-pin card-edge connector allows connection to other parts of the DA&C system. The panels are compatible with the PCI-20013A ground-plane cables (See Note 1 in the Specifications). PCI-20018T-1 mounts in the PCI-20029A rack mount enclosures. PCI-20048T-1 fits into the PCI-20051A-1 enclosure, which has an optional cover, the PCI-20052A-1. Figure 1 shows the physical layout of the PCI-20048T-1. The PCI-20018T-1 is the same except for being only eight inches long (20cm).

SPECIFICATIONS—PCI-20018T-1, PCI-20048T-1

All specifications are typical at +25°C, unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Function	Isolated, Digital, Input/Output, Signal Conditioning Termination Panel	
I/O Configuration	PCI-20018T-1 PCI-20048T-1	8 channels(1) 16 channels
Terminal/Connections		
Field Wires	Terminal Type Wire Size Range Number of Terminals PCI-20018T-1 PCI-20048T-1	#6 screws 22-12 AWG 18 34
System Connection	50-Pin Mating Connector	Ansley #609-5015M
Power Available	System Power, fused at 0.25A	Via ribbon cable(2)
Size	Length × Width × Height(3) PCI-20018T-1 PCI-20048T-1	 8" × 3.5" × 2.1" 20.3cm × 8.9cm × 5.3cm 14" × 3.5" × 2.1" 35.6cm × 8.9cm × 5.3cm
Temperature Range	Panel Temperature	−30 to +70°C

NOTES: (1) Each digital I/O port in the PCI-20000 system supports 16 channels. Therefore, when a cable such as the PCI-20013A series is connected to a PCI-20018T-1, which accommodates eight channels, the other eight channels in that cable are not available for use. (2) When used with PCI-20000 Carriers or Modules, the System connection cable brings +5V from the computer supply to the termination panel. This is used to power on board functions including the LEDs. (3) This height includes the height of the PCI-1100 series modules. The height of the board itself is 1.4" (3.6cm).

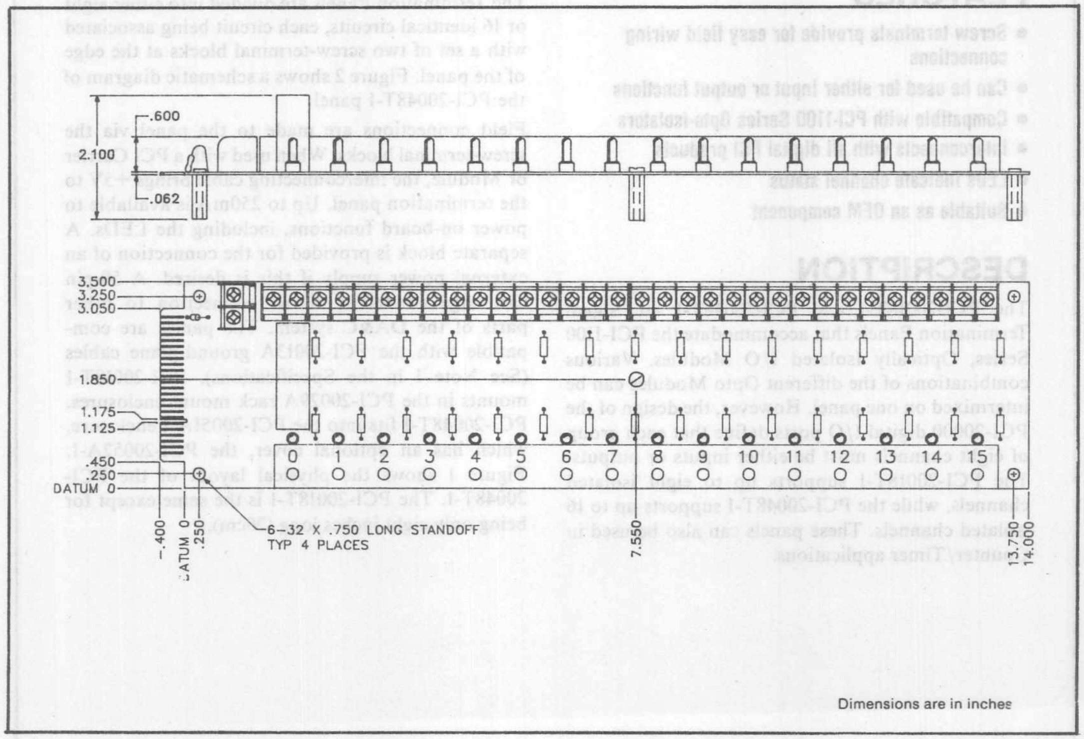


FIGURE 1. PCI-20048T-1 Mechanical Layout.

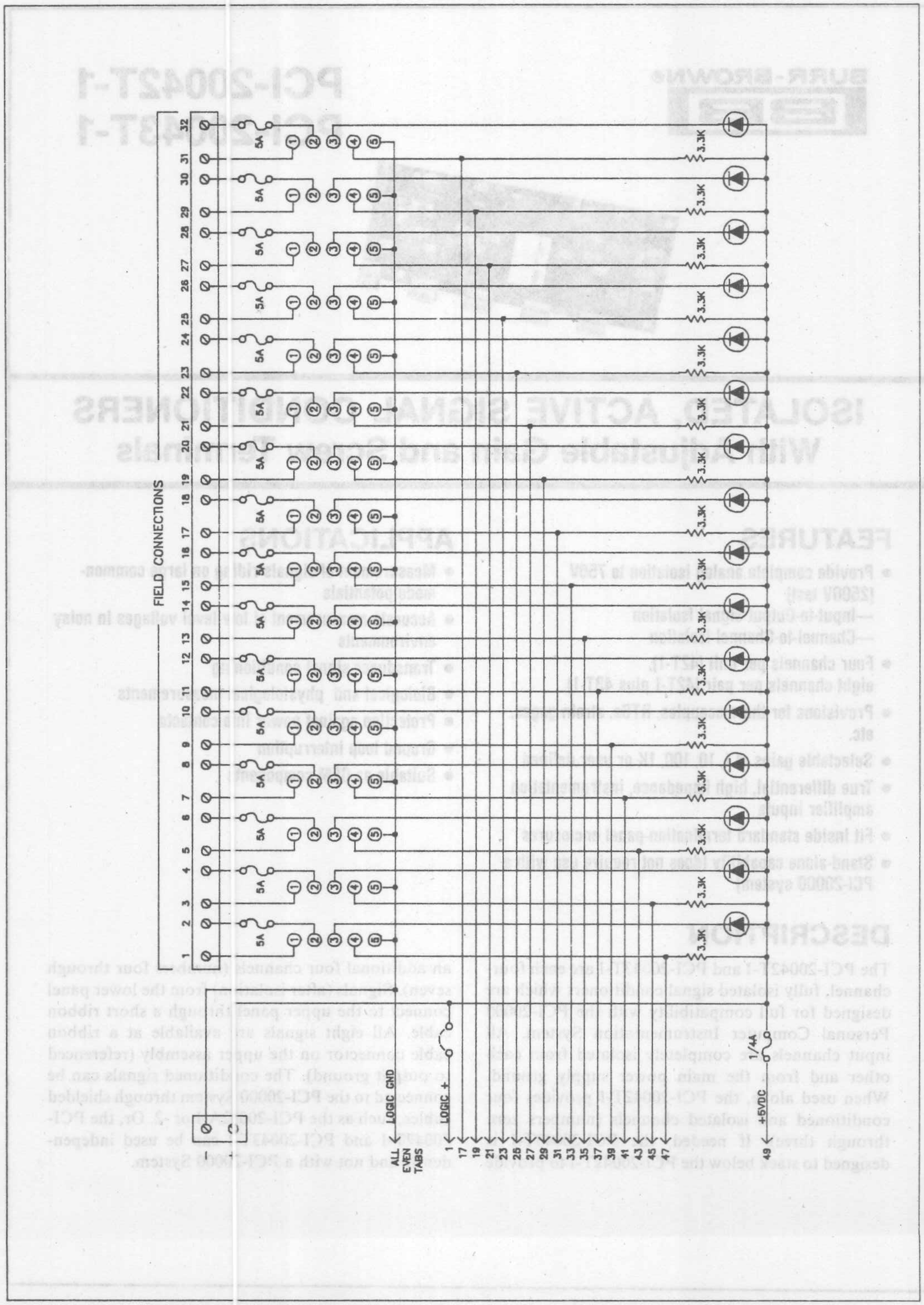
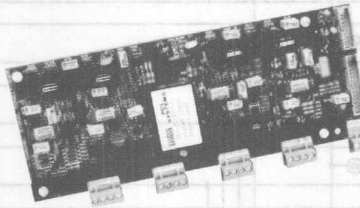


FIGURE 2. PCI-20048T-1 Schematic Diagram.

BURR-BROWN®

PCI-20042T-1
PCI-20043T-1



ISOLATED, ACTIVE SIGNAL CONDITIONERS
With Adjustable Gain and Screw Terminals

FEATURES

- Provide complete analog isolation to 750V (2500V test)
  —Input-to-Output signal isolation
  —Channel-to-Channel isolation
- Four channels per unit (42T-1), eight channels per pair (42T-1 plus 43T-1)
- Provisions for thermocouples, RTDs, strain gages, etc.
- Selectable gains of 1, 10, 100, 1K or user defined
- True differential, high impedance, instrumentation amplifier inputs
- Fit inside standard termination-panel enclosures
- Stand-alone capability (does not require use with a PCI-20000 system)

APPLICATIONS

- Measurement of signals riding on large common-mode potentials
- Accurate measurement of low-level voltages in noisy environments
- Transducer signal conditioning
- Biological and physiological measurements
- Protection against power line contacts
- Ground loop interruption
- Suitable as OEM components

DESCRIPTION

The PCI-20042T-1 and PCI-20043T-1 are each four-channel, fully isolated signal conditioners which are designed for full compatibility with the PCI-20000 Personal Computer Instrumentation System. All input channels are completely isolated from each other and from the main power supply ground. When used alone, the PCI-20042T-1 provides four conditioned and isolated channels (numbers zero through three). If needed, the PCI-20043T-1 is designed to stack below the PCI-20042T-1 to provide an additional four channels (numbers four through seven). Signals (after isolation) from the lower panel connect to the upper panel through a short ribbon cable. All eight signals are available at a ribbon cable connector on the upper assembly (referenced to output ground). The conditioned signals can be connected to the PCI-20000 System through shielded cables, such as the PCI-20012A-1 or -2. Or, the PCI-20042T-1 and PCI-20043T-1 can be used independently and not with a PCI-20000 System.

Input signals are amplified by a true differential-input instrumentation amplifier. The low noise, excellent DC stability, and low nonlinearity preserve the integrity of low-level signals. These features can be critical in many applications. The amplified input signals are passed to a high performance isolation amplifier which translates the input signals so that they are referenced to the output ground. Any input common-mode voltages that exist with respect to output ground are rejected by the isolation barrier. This allows the interruption of ground loops that would otherwise lead to serious system errors. Isolation permits the measurement of small signals in the presence of large common-mode voltages, while protecting other connected instrumentation from such voltages.

These Isolation Panels can accommodate input signals from transducers such as thermocouples, RTDs, and strain gages. In addition to amplification and isolation, the panels have provisions for specialized types of signal ocnditioning. An on-board cold junction compensation network allows any mixture of thermocouple types. For bridge configurations, each channel includes a constant current source for bridge excitation and mounting locations for user-installed bridge completion resistors. Other locations allow for user-installed components that permit the incorporation of one or two poles of filtering, voltage dividers, input protection, etc.

On-board calibration potentiometers allow the user to null input offsets and adjust gain and excitation currents if required.

Input power for each of the signal conditioning panels comes from an external ±15VDC supply. Both the PCI-20042T-1 and the PCI-20043T-1 include a DC-to-DC Converter (power supply) which provides isolated DC power for the four channels of amplification and signal conditioning on each panel.

The PCI-20038A Series Power Supply is available for powering the PCI-20042T and PCI-20043T Isolation Panels and is capable of handling up to eight each of the PCI-20042T and PCI-20043T (64 isolated channels).

The PCI-20029A enclosure is available to house these Isolation Panels in a table-top or rack-mount environment. Each enclosure will house up to 32 channels consisting of PCI-20042T-1 and PCI-20043T-1 Isolation Panels.

User documentation shipped with these Isolation Panels provides complete instructions for setting gains, adding signal conditioning components, and making calibration adjustments.

SPECIFICATIONS—PCI-20042T-1, PCI-20043T-1

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		PCI and VME Systems
I/O Configuration	Differential	Analog Input
Number of Channels	PCI-20042T-1	Four
	PCI-20042T-1 and PCI-20043T-1	Eight
Isolation		In/Out and Channel/Channel
Excitation	For resistive applications	Current Source
Cold Junction Compensation	For thermocouples	Si Sensor
Input Stage		
Gain: Jumper[1]	Programmable Gain	G = 1, 10, 100, 1K
User	Defined via R_G, G_A = 1 or 10	G = 1 to 1K
Equation[2]	Accuracy = ±20%	$1 + [40K / (R_G + 40)]\ G_A$
Gain: Accuracy		Trimmable to ±0.1%
Drift		320ppm/°C
Linearity	G = 1, 10	±0.083%FS
	G = 100	±0.083%FS
	G = 1000	±0.096%FS
Offset: Voltage		Trimmable to zero ±1LSB
Drift	RTI	1mV/°C max
Common-Mode: Rejection	60Hz, 1K unbalanced	
	G = 1	80dB
	G = 10 to 1K	96dB
Range		10V, DC + peak AC
Bias Current		6nA
	vs Temperature	.1nA/°C
Offset Current		3nA
	vs Temperature	.1nA/°C
Input Impedance		100MΩ at 15pF
Crosstalk	Channel-to-channel	−100dB
Output Stage		
Offset: Voltage	RTO	Trimmable to zero
Drift	RTO	±.5mV/°C
Current		1mA
Frequency Response	Full Scale, to −3dB:	
	G = 1, 10, 100, 1000	100Hz
Slew Rate	G = 1 to 100	.15V/μs
Settling Time	To .1%, 10V Step:	
	G = 1, 10	2 millisec
	G = 100	2 millisec
	G = 1000	2.4 millisec
Isolation Ratings		
Voltage	Continuous, DC + Peak AC	750V
	Test, 10 Seconds	2500V
Leakage Current	240Vrms, 60Hz	1μA
Isolation Mode Rejection	60Hz, G = 1000	127dB
Excitation Current	For resistive devices	
Adjustment Range		1.1mA to 2mA
Factory Setting		1.4mA ±2μA
Drift	I_{OUT} = 1mA	−.25%/°C
	I_{OUT} = 1.5mA	±.01%/°C
	I_{OUT} = 2mA	±.08%/°C
Compliance		14V
Cold-Junction Sensor	Thermocouple Compensation	
Scale Factor		10mV/°K, ±1°K
Power Requirements	Eight Channels (42T + 43T)	±15V at ±110mA
Size[3]	Length × Width × Height	9″ × 3.5″ × 1.4″ 22.9cm × 8.9cm × 3.6cm
Temperature Range	Board Temperature	0 to +70°C

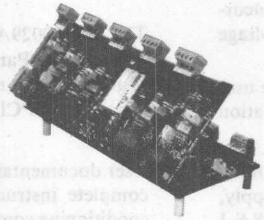
NOTES:

(1) Overall gains of 1 to 1K are produced by combinations of first and second stage gains of 1, 10, 100 and 1 and 10 respectively.

(2) G_A is the second stage gain (isolation amplifier).

(3) For one panel only. Please refer to the PCI-20044T-1 Data Sheet, Figure 1, for the size of the stacked configuration.

BURR-BROWN®



PCI-20044T-1
PCI-20045T-1

ACTIVE SIGNAL CONDITIONERS
With Adjustable Gain, Transducer Excitation, and Screw Terminals

FEATURES

- TRUE DIFFERENTIAL AMPLIFIER FOR EACH CHANNEL
- FOUR CHANNELS PER UNIT (44T-1), EIGHT CHANNELS PER PAIR (44T-1 plus 45T-1)
- SELECTABLE GAINS OF 1, 10, 100, 1K OR USER DEFINED
- CURRENT SOURCE EXCITATION FOR BRIDGE TYPE TRANSDUCERS
- COLD JUNCTION COMPENSATION FOR THERMOCOUPLES
- STAND-ALONE CAPABILITY (does not require use with a PCI-20000 system)
- FITS INSIDE STANDARD TERMINATION PANEL ENCLOSURES

APPLICATIONS

- ACCURATE MEASUREMENT OF LOW-LEVEL VOLTAGES IN NOISY ENVIRONMENTS
- TRANSDUCER SIGNAL CONDITIONING
- BIOLOGICAL AND PHYSIOLOGICAL MEASUREMENTS
- SUITABLE AS AN OEM COMPONENT

DESCRIPTION

The PCI-20044T-1 and PCI-20045T-1 are four-channel signal conditioners which are designed for full compatiblity with the PCI-20000 Personal Computer Instrumentation System. Each input channel's gain is adjustable, independent from all other channels. When used alone, the PCI-20044T-1 provides four conditioned channels (channel numbers zero through three). If needed, the PCI-20045T-1 is designed to stack below the PCI-20044T-1 to provide an additional four channels (channel numbers four through seven). Signals from the lower panel connect to the upper panel through a short ribbon cable. All eight signals are available at a ribbon cable connector on the upper assembly (referenced to output ground). The conditioned signals can be connected to the PCI-20000 System through shielded cables, such as the PCI-20012A-1 or -2.

Input signals are amplified by a true differential-input instrumentation amplifier whose low noise, excellent DC stability, and low nonlinearity preserve the low-level signals that are critical to many applications. The differential input of this amplifier provides excellent rejection of extraneous common-mode voltages that may exist with respect to the input reference (ground) points.

These Termination Panels can accommodate input signals from transducers such as thermocouples, RTDs and strain gages. In addition to amplification, the panels have provisions for specialized types of signal conditioning. An on-board cold junction compensation network allows any mixture of thermocouple types. For bridge configurations, each channel includes a constant current source for bridge

excitation and mounting locations for user-installed bridge completion resistors. Other locations allow for user-installed components that permit the incorporation of one or two poles of filtering, voltage dividers, input protection, etc.

On-board calibration potentiometers allow the user to null input offsets and adjust gain and excitation currents if required.

Input power for each of the signal conditioning panels comes from an external line-voltage supply, such as the PCI-20038A-1. The PCI-20038A-1 115VAC Power Supply is capable of handling up to 24 each of the PCI-20044T-1 and PCI-20045T-1 (192 channels). The PCI-20038A-3 is a similar supply designed for 240VAC nominal line voltage.

The PCI-20029A-1 enclosure is available to house these Termination Panels in a table-top or rack-mount environment. Each enclosure will house up to 32 channels consisting of PCI-20044T-1 and PCI-20045T-1 Panels.

User documentation shipped with these Panels provides complete instructions for setting gains, adding signal conditioning components, and making calibration adjustments.

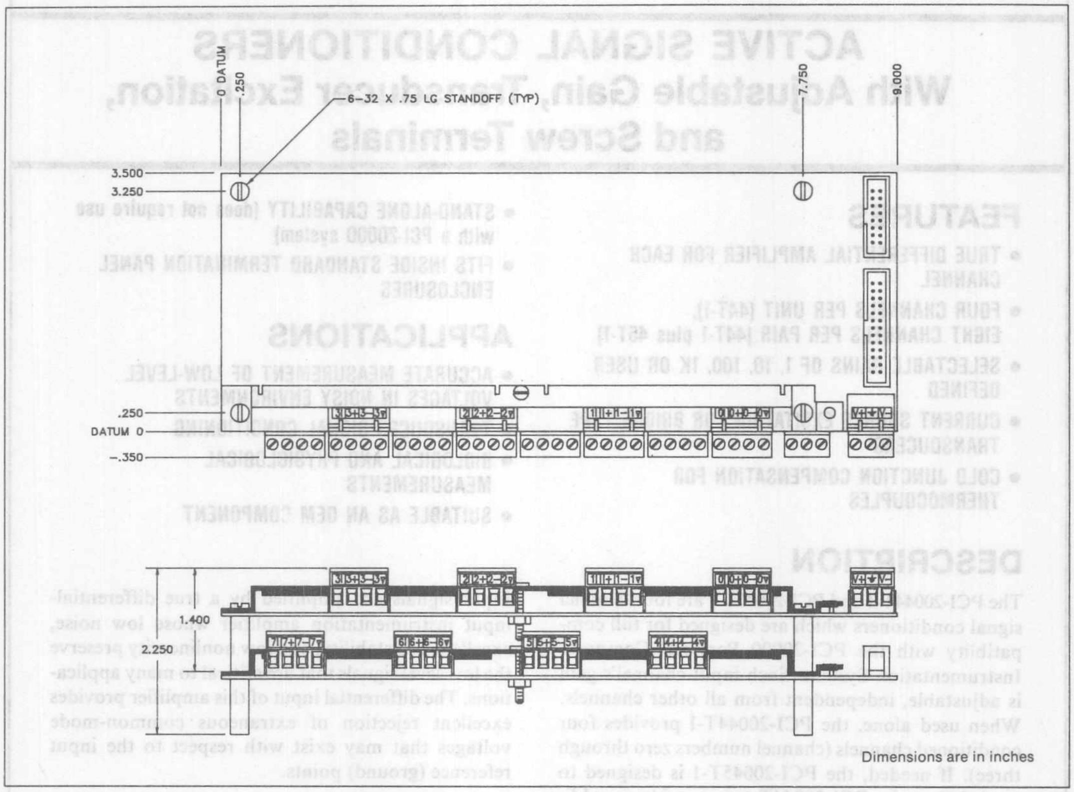


FIGURE 1. PCI-20044T/45T Signal Conditioner Termination Panel/Expander Dimensions and Mounting Details.

SPECIFICATIONS—PCI-20044T-1, PCI-20045T-1

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Compatibility		PCI and VME Systems
I/O Configuration	Differential	Analog Input
Number of Channels	PCI-20044T-1	4
	PCI-20044T-1 and PCI-20045T-1	8
Excitation	For resistive applications	Current Source
Cold Junction Compensation	For thermocouples	Si Sensor
INPUT STAGE		
Gain: Jumper	Programmable Gain	G = 1, 10, 100, 1K
User	Defined via R_G	G = 1 to 1K
Equation	Accuracy = ±20%	$1 + [40K/(R_G + 40)]$
Gain Accuracy	G = 1, 10	±0.05%FS
	G = 100	±0.15%FS
	G = 1000	±0.5%FS
Drift	G = 1	5ppm/°C
	G = 10	10ppm/°C
	G = 100	15ppm/°C
	G = 1000	20ppm/°C
Linearity	G = 1, 10	±0.01%
	G = 100	±0.02%
	G = 1000	±0.05%
Offset Voltage		Trimmable to zero
Drift	RTI	7μV/°C
Common-Mode Rejection	60Hz, 1KΩ unbalanced	
	G = 1	80dB
	G = 10 to 1K	96dB
CM Range		10V, DC + peak AC
Bias Current		6nA
	vs Temperature	0.1nA/°C
Offset Current		3nA
	vs Temperature	0.1nA/°C
Input Impedance		100MΩ at 15pF
OUTPUT STAGE		
Current		1mA
Frequency Response	Full Scale, to −3dB	
	G = 1	30kHz
	G = 10	3kHz
	G = 100	300Hz
	G = 1000	30Hz
Slew Rate		0.2V/μs
Settling Time	to 0.01%, 10V Step	
	G = 1, 10	60μs
	G = 100	500μs
	G = 1000	4500μs
Excitation Current	For Resistive Devices	
Adjustment Range		1.1mA to 2mA
Factory Setting		1.4mA ±2μA
Drift	I_{OUT} = 1mA	−0.25%/°C
	I_{OUT} = 1.5mA	+0.01%/°C
	I_{OUT} = 2mA	+0.08%/°C
Compliance		14V
Cold-Junction Sensor	Thermocouple Compensation	
Scale Factor		10mV/°K, ±1°K
Power Requirements	Four Channels	±15V at 20mA
Physical Size(1)	Length × Width × Height	9" × 3.5" × 1.4" 22.9cm × 8.9cm × 3.6cm
Temperature Range	Board Temperature	0 to +70°C

NOTES: (1) For one panel only. See Figure 1 for the size of the stacked configuration.

BURR-BROWN®

PCI-1101
PCI-1102
PCI-1103
PCI-1104
PCI-1105
PCI-1106

DIGITAL OPTO-ISOLATOR MODULES

FEATURES

- **Compatible with PCI-20018T and PCI-20048T Termination Panels**
- **Models available for both input and output functions**
- **Converts a wide range of voltages to TTL levels**
- **Switches up to 60VDC and 240VAC at 3A**

DESCRIPTION

The PCI-1100 Series are Digital Opto-Isolation Modules that plug into the PCI-20018T and PCI-20048T Digital Signal Termination Panels. One Module is required for each channel. Two types, input and output, are available in six different models for a wide range of applications. All input models accept both AC and DC voltages. Each provides a TTL output to drive a standard PCI digital input. Different models are provided for AC and DC outputs. These units convert TTL outputs from the PCI system to switch higher voltage (and current) loads. Figure 1 shows simplified diagrams for the various types. Note that the DC output device (PCI-1103) presents an open collector NPN transistor to the load. The AC output units (PCI-1104 and PCI-1106) contain zero crossing circuitry and switch their loads with triacs. When an Opto Module senses a valid input, it lights an LED on the Termination Panel as an indicator.

SPECIFICATIONS—PCI-1101 thru PCI-1106

All specifications are typical at +25°C unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Function	Digital input/output isolation	
Compatibility	Termination panels	PCI-20018T and 48T
I/O Configuration		1 Channel/Module
Input Types	AC or DC inputs	PCI-1101, 02, 05
Voltage Range		
On V/I	PCI-1101, AC input	15–32V/12–30mA
	DC input	10–32V/8–30mA
	PCI-1102, AC/DC input	90–140V/6–10mA
	PCI-1105, AC/DC input	180–280V/4–7mA
Off V/I	PCI-1101	3V/1mA
	PCI-1102	45V/3mA
	PCI-1105	80V/1mA
Isolation	Input to output	4000V
Impedance		$10^{10}\Omega$ at 8pF
Output Levels	3.3K pull-up, 50mA pull-down	TTL
Turn-On Time	PCI-1101	5ms max
	PCI-1102, 1105	20ms max
Turn-Off Time	PCI-1101	5ms max
	PCI-1102, 1105	20ms max
Output Type	DC loads	PCI-1103
Load Range, V/I	at 45°C (at 70°C)	5–60V/3A (2A)
Minimum Load Current		20mA
Voltage Drop	Across output transistor	1.6V max
Off Leakage	at 60V	1mA
Isolation	Input to output	4000V rms
Impedance		$10^{10}\Omega$ at 8pF
Input Levels		TTL
Turn-On Time		100μs
Turn-Off Time		750μs
Output Type	AC Loads	PCI-1104, PCI-1106
Load Range, V/I	PCI-1104, at +45°C (at +70°C)	12–140V/3A (2A)
	PCI-1106, at +45°C (at +70°C)	24–280V/3A (2A)
Minimum Load Current		20mA
Voltage Drop	Across output triac	1.6V max
Frequency Range		25–65Hz
Off Leakage	at 120V, 60Hz	5mA rms
Isolation	Input to output	4000V rms
Impedance		$10^{10}\Omega$ at 8pF
Input Levels		TTL
Turn-On Time		1/2 cycle max
Turn-Off Time		1/2 cycle max
Size	Length × Width × Height	1.7″ × 0.6″ × 1.25″ 4.3cm × 1.5cm × 3.2cm
Temperature Range	Module temperature	−30 to 70°C

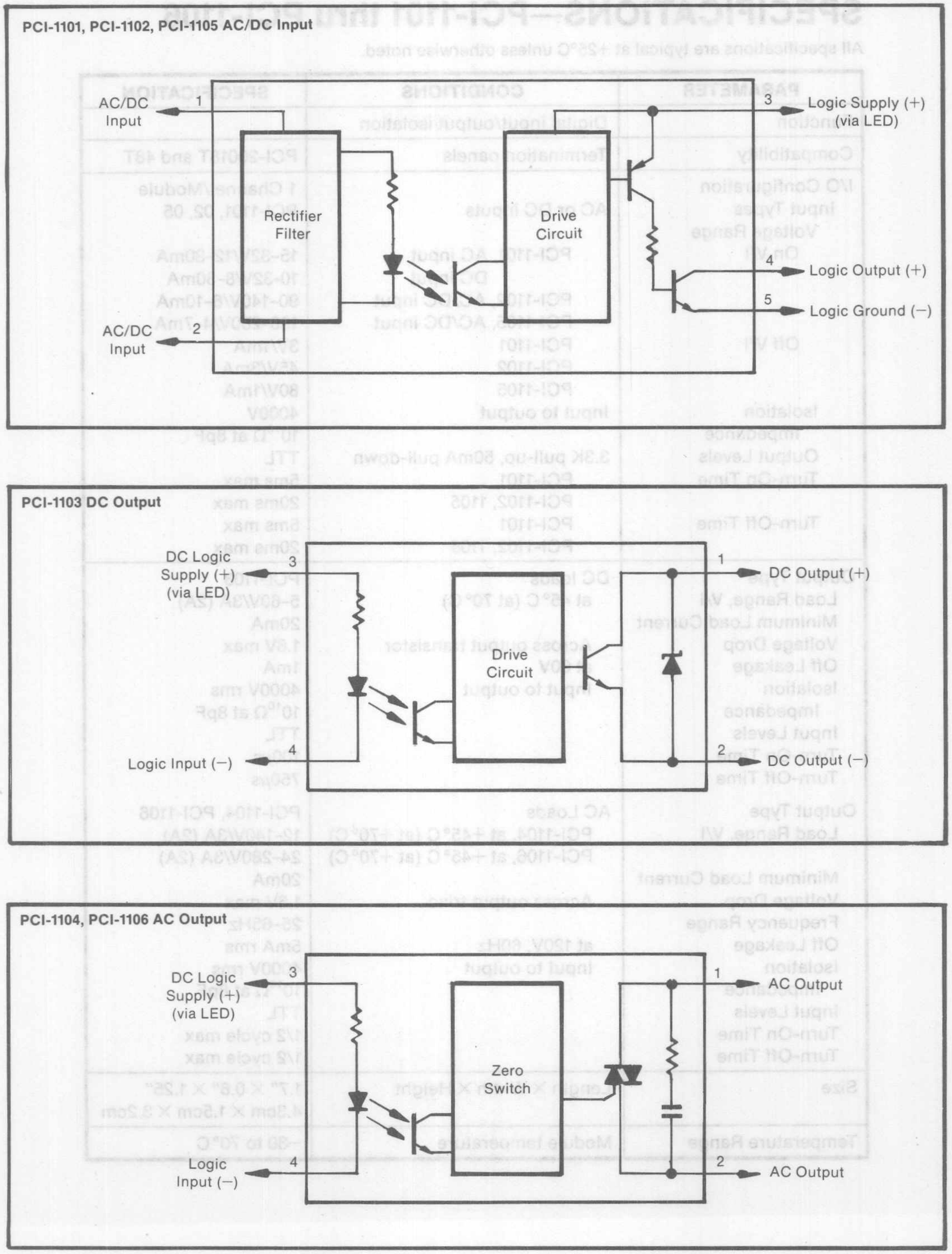
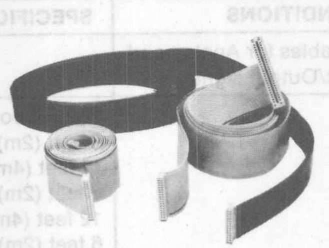


FIGURE 1. Opto-Isolated Modules.

BURR-BROWN®

PCI-20012A-1
PCI-20012A-2
PCI-20013A-1
PCI-20013A-2
PCI-20032A-1
PCI-20061A-1



CABLES

FEATURES

- Flat ribbon cable with connectors on both ends
- Most cables are available in six-foot (2m) and 12-foot (4m) lengths
- Can be used for either input or output (except PCI-20032A-1)
- Compatible with all PCI and selected VME products
- Suitable as OEM components

DESCRIPTION

The PCI-20012A series are fully shielded flat ribbon cables intended for either analog input or output use. The shield minimizes both noise pickup and emission. The cables contain 26 wires and are terminated with T&B Ansley (or equivalent) female connectors. PCI-20012A-1 is six feet (2m) long, while PCI-20012A-2 is 12 feet (4m) long. All shielded cables have the shield connected at one end only to avoid ground loops. The grounded end is clearly marked.

The PCI-20013A series and the PCI-20061A-1 are built with 'ground-plane' type flat ribbon cable. They are intended for either digital input or output use. Included in this category are the Counter/Timer applications. The ground-plane minimizes cable inductance while reducing electrostatic and electromagnetic emissions. The cables contain 34 wires and are terminated with T&B Ansley (or equivalent) female connectors. The PCI-20013A-1 and the PCI-20061A-1 are six feet (2m) long, while the PCI-20013A-2 is 12 feet (4m) long. All ground-plane cables have the the ground plane connected at one end only to avoid ground loops. The grounded end is clearly marked.

The PCI-20032A-1 is a special purpose, multi-connector cable intended for analog output use. It is built with non-shielded flat ribbon cable. One end of the cable is intended to connect to the PCI-20010T-1 or PCI-20057T-1 Termination Panels. The other end actually has three connectors that are intended to plug on to three different, one or two channel, analog output modules (PCI-2003M-2, PCI-20003M-4, PCI-20006M-1, or PCI-20006M-2). This has the effect of saving space and reducing cost by connecting up to six analog outputs to one Termination Panel instead of to three.

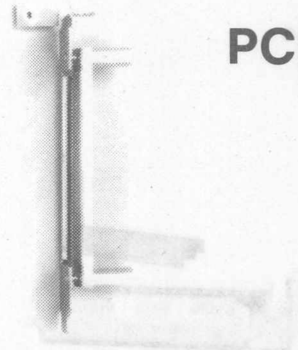
SPECIFICATIONS

PCI-20012A-1, PCI-20012A-2, PCI-20013A-1, PCI-20013A-2, PCI-20032A-1, PCI-20061A-1

PARAMETER	CONDITIONS	SPECIFICATION
Function	Connecting Cables for Analog and Digital Input/Output signals	
Configuration(1)		Flat Ribbon Cable
Length	PCI-20012A-1	6 feet (2m)
	PCI-20012A-2	12 feet (4m)
	PCI-20013A-1	6 feet (2m)
	PCI-20013A-2	12 feet (4m)
	PCI-20032A-1	6 feet (2m)
	PCI-20061A-1	6 feet (2m)
Shield Type(2)	PCI-20012A-1/-2	Full
	PCI-20013A-1/-2	Ground-Plane
	PCI-20032A-1	None
	PCI-20061A-1	Ground-Plane
Number of Wires	PCI-20012A-1/-2, 32A-1	26
	PCI-20013A-1/-2, 61A-1	34
Wire Size	All Cables	28 gage stranded
Connectors(3)	PCI-20012A-1/-2, Cable ends	609-2630
	PCI-20013A-1/-2, Cable end(1)	609-3430
	Cable end(2)	609-5015M
	PCI-20032A-1, Cable ends	609-2630
	PCI-20061A-1, Cable ends	609-3430
Temperature Range		0 to +70°C

NOTES: (1) Various cable and connector manufacturers place marks or codes to indicate the location of wire or pin number 1. When using ANY PCI system, these codes or marks should be IGNORED. The correct wire and pin designations are described in the PCI users manuals. (2) Both shielded and ground-plane cables have the shield connected at one end only to avoid ground loops. The shield-connected end is marked 'Computer.' (3) The part numbers shown are 'T&B Ansley' numbers. They are intended for reference only, as several second sources are available.

BURR-BROWN®



PCI-20028A-3

STRAIN-RELIEF BRACKETS

FEATURES

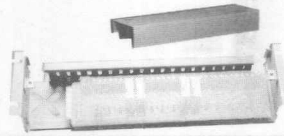
- Supports cables as they exit from the PC
- Compatible with all PCI series ribbon cables
- Works with IBM PC, XT, AT and other hardware compatible PCs
- Suitable as an OEM component

DESCRIPTION

The PCI-20028A-3 is a metal bracket designed to support flat ribbon cables as they exit from the personal computer. The bracket fits into the rear of an expansion board slot, where a board hold-down device would otherwise go. An adjustable clamp allows from one to five cables to be accommodated. Use of this bracket helps prevent damage to the PC or installed boards if the cables are accidentally pulled on. The PCI-20028A-3 is installed adjacent to either a PCI-20001C or PCI-20041C Carrier in an unused slot of the computer. In most cases one PCI-20028A-3 bracket is recommended for each Carrier (in addition to the combination clamp that is automatically supplied) if one or more instrument modules will be plugged into the Carrier.

BURR-BROWN®

PCI-20029A-1
PCI-20051A-1
PCI-20052A-1



TERMINATION PANEL ENCLOSURES

FEATURES

- **Holds from one to four PCI style Termination Panels**
- **Mounts in a standard 19-inch (48cm) rack**
- **Provides both mechanical and electrical protection**
- **Suitable as an OEM component**

DESCRIPTION

The PCI-20029A-1 is a metal enclosure designed to support up to four standard-size PCI Termination Panels. All PCI-20000 models are accommodated except the PCI-20048T (which fits into the PCI-20051A-1). The PCI-20029A-1 is a general-purpose enclosure intended for both analog and digital panels.

All enclosures mount into standard 19-inch (48cm) relay racks, but can also be used in a table-top configuration. A cover is included with the PCI-20029A-1. An optional cover, the PCI-20052A-1, is available for the PCI-20051A-1.

These enclosures help prevent mechanical damage to the Termination Panels while restricting accidental human contact with internal circuits. A plastic cable tray in the PCI-20051A-1 keeps the field wiring neat. Figures 1 and 2 suggest the mechanical configurations and give the major dimensions.

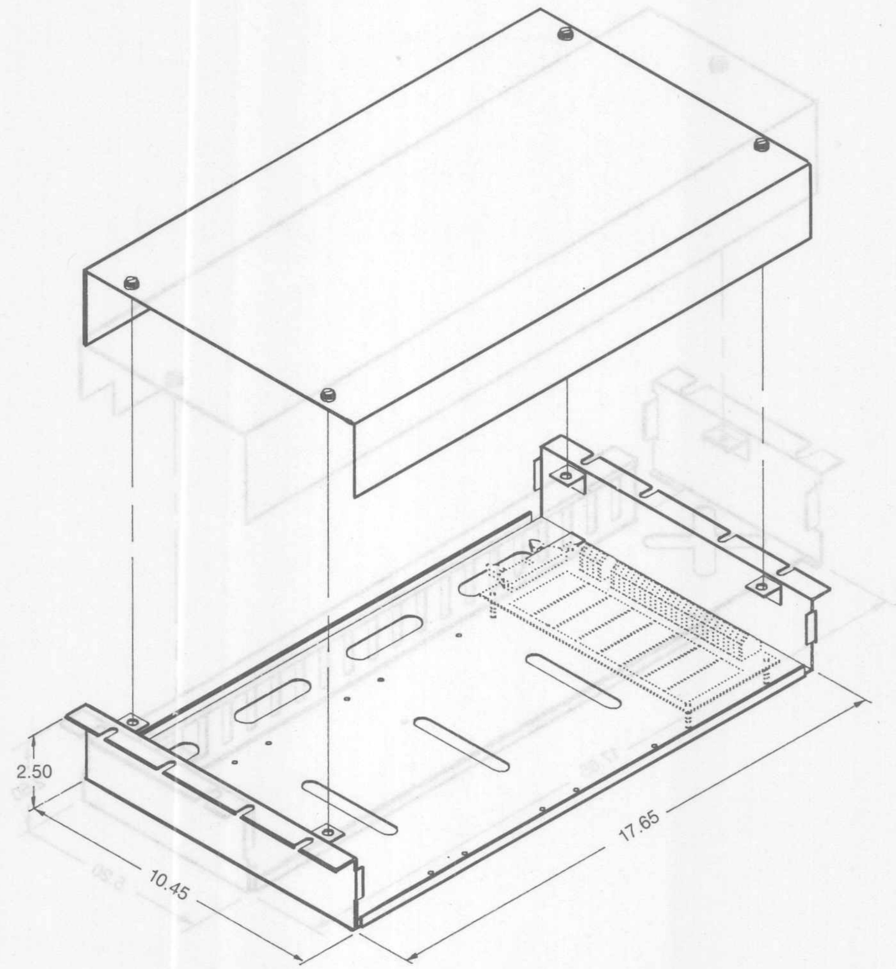


FIGURE 1. PCI-20029A-1 Enclosure.

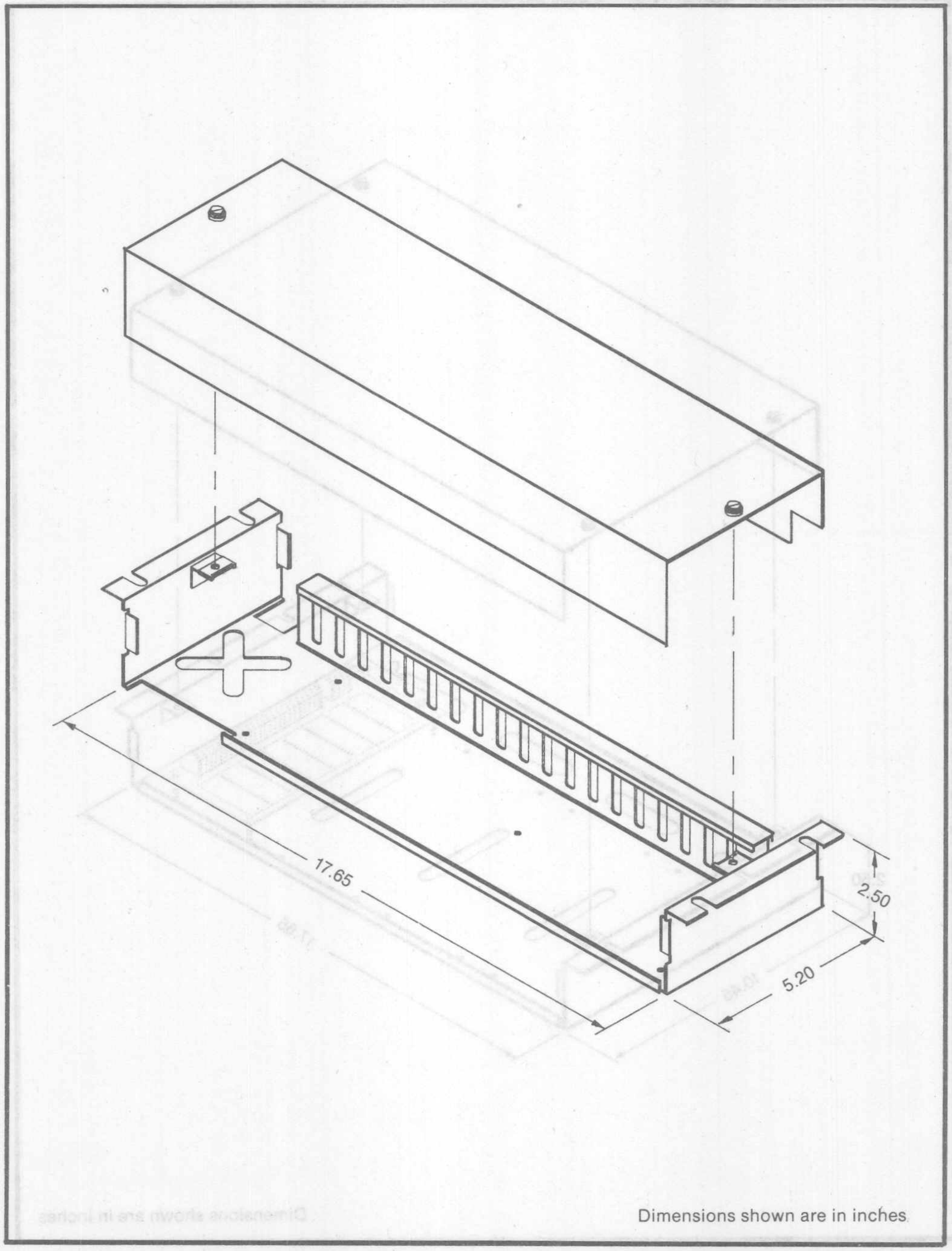
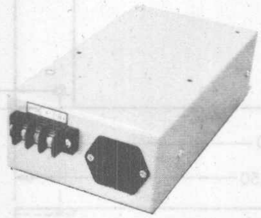


FIGURE 2. PCI-20051A-1 Enclosure and PCI-20052A-1 Cover.

BURR-BROWN®

PCI-20038A-1 (115 VAC)
PCI-20038A-3 (240VAC)



±15 VOLT DC POWER SUPPLY

FEATURES

- ±15VDC at 0.8A
- ±0.05% line and load regulation
- Short-circuit protected
- Convenient plug and terminal connections

DESCRIPTION

The PCI-20038A series consists of general purpose ±15VDC power supplies. They are available in both 115V and 240VAC line models. Each has adjustable outputs and is capable of delivering 800mA to its loads. The supplies are completely enclosed in a metal case, offering both mechanical and electrical contact protection. While they can be used for almost any purpose, the PCI-20038A is primarily intended to power the PCI-20042T thru PCI-20045T series of Active Signal Conditioners. If desired, the supplies can be mounted either inside or on the rear of the PCI-20029A Termination Panel Enclosures.

Figure 1 shows the physical dimensions of the power supplies, and the input and output connectors.

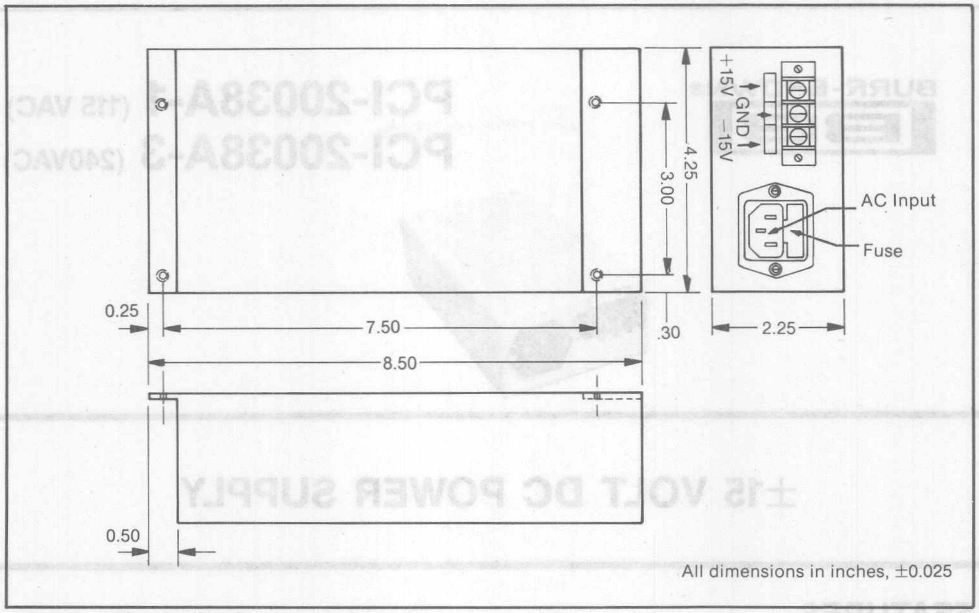


FIGURE 1. PCI-20038A ±15VDC Power Supply.

SPECIFICATIONS—PCI-20038A-1, PCI-20038A-3

All specifications are typical at +25°C, unless otherwise noted.

PARAMETER	CONDITIONS	SPECIFICATION
Function		DC Power Supply
AC Input: Voltage	PCI-20038A-1 PCI-20038A-3	105-132VAC 209-264VAC
Frequency		47-63Hz
Fuse	Style PCI-20038A-1 PCI-20038A-3	5 × 20mm, slow-blow 0.5A 0.25A
DC Output Voltage:		±15V
Adjustment Range		±5%
Load Current		±800mA
Regulation	Line, for a 10% line change Load, for a 50% load change	±0.05% ±0.05%
Output Ripple	Peak to Peak, Maximum	5mV
Output Protection		Current limit/Foldback
Efficiency		55%
Terminal/Connections:		
Load Wires	Terminal type Configuration	#6 screw +15V, Ground, −15V
Line Connection		Standard AC Socket
Size	Length × Width × Height	9.2″ × 4.2″ × 2.3″ 23.4 × 10.7 × 5.8cm
Weight		4lb (1.8kg)
Mounting	Via four threaded holes	#6-32
Temperature Range		0 to 50°C

Section 11

PERSONAL-COMPUTER SOFTWARE

FOR

DATA ACQUISITION, ANALYSIS, TEST, MEASUREMENT, AND CONTROL

SOFTWARE INTRODUCTION

11

TABLE OF CONTENTS

*NOTE: The PCI-20046S Series of software supersedes the original PCI-20014S Series.

POWERFUL SOFTWARE FOR PCI SYSTEMS

THE PCI-20000 SYSTEM

The PCI-20000 hardware system for data acquisition, test, measurement, and control consists of "carrier" boards which plug into IBM PC, XT, AT, and compatible computer expansion slots, (or into slots in an expansion chassis). These carriers in turn accommodate up to three "Instrument Modules" each, that plug "piggy-back" into the carrier. This provides the user with a modular system of real-world I/O that is extremely flexible.

Any combination of Instrument Modules can be selected to meet specific application requirements. This is a powerful approach that allows standard Instrument Modules to be specified and configured by the user into a system optimized for his individual requirements. This maximizes performance and minimizes cost, since the user needs to buy only those components required to get his job done satisfactorily.

As requirements change, different Instrument Modules can be substituted to solve a new problem. More Instrument Modules can be added if requirements expand. As new modules are added to the product line, performance can be upgraded easily.

PCI-20000 SOFTWARE

There is a wide range of software packages for the PCI-20000. These are summarized immediately below. More detailed data sheets appear later, after this summary. The PCI-20000 family of software supports all analog I/O (including thermocouples), digital I/O, and counter-timer functions. Four major categories exist:

- General-purpose software.
- High-performance software.
- Third-party software offered by other companies.
- Demonstration diskettes for some of the above software and hardware.

GENERAL-PURPOSE SOFTWARE

- BASIC Language Interface (PCI-20046S-1). BASIC is the most popular of all personal computer languages. It is easy to learn and easy to use.
- C Language Interface (PCI-20056S-2). C is an efficient high-level language that allows high-speed operations. The PCI-20046S-2 supports Lattice or Microsoft C compilers (through compiler version 3.X).
- Turbo Pascal Language Interface (PCI-20046S-3). Pascal is a structured language for organized programming, and it also provides high efficiency. Turbo Pascal is a version of Pascal published by Borland International. It is fast and is very easy to learn.
- ASYST Data Acquisition Interface (PCI-20046S-4). ASYST itself is a scientific software package available from ASYST Software Technology. It normally consists of three "Modules." Modules 1 and 2 are used for analysis and graphics. The ASYST Data Acquisition Interface Module (PCI-20046S-4) substitutes for ASYST Module 3 and provides the interface between the PCI-20000 system hardware and ASYST modules 1 and 2.

  When used with the PCI-20046S-4, ASYST Modules 1 and 2:

  1) Allow direct data acquisition from PCI-20000 series Instrument Modules;
  2) Easily reduce, manipulate and analyze acquired data;
  3) Produce high-quality scientific graphics;
  4) Compare results at a glance using multiple-window displays;
  5) Integrate analysis functions, including Fourier analysis, with graphics.
- Combination Software Package (PCI-20046S-6). A combination of the first three software packages listed above (one each: PCI-20046S-1, PCI-20046S-2, PCI-20046S-3). This package is for the user who wants to add multiple-language programming capability to his PCI-20000 system at a greatly reduced cost compared to buying each language-package separately.

Each of the PCI-20046S Interface Packages includes these features:

- Optimized I/O Routines. Internal routines are written in assembly language to provide maximum speed. Special data acquisition routines allow high data rates independent of the programming language used.
- Assembly Language. Each package includes detailed instructions showing how to access the assembly language routines directly. This is for the user who prefers to work in assembly language to obtain increased speed and efficiency.
- Thermocouple Interfacing. Linearization and cold-junction compensation for J, K, and T thermocouples are included in each interface package at no extra cost to the user! Analog readings may be returned directly in degrees, or the user may access the linearization and compensation routines to post-process his data.
- Tutorial and Error Checking. Each of the PCI-20046S language support packages provides tutorial and demonstration material to aid a new user in getting started with the PCI-20000 system. Error-checking capability is also included in each package with appropriate error codes to describe difficulties encountered.

HIGH-PERFORMANCE SOFTWARE

- High-Speed Interface (PCI-20047S-1). Interfaces with each of the general-purpose PCI-20046S Series to greatly increase the speed of the PCI-20000 System. It also provides DMA capability when used with the PCI-20041C-3 carrier.

HIGH PERFORMANCE SOFTWARE (Continued)

- PCI ControLOGraph (PCI-20056K-1) Intelligent Data Logger System. While not just a software package alone, the PCI ControLOGraph System contains powerful software that provides traditional data-logger functions, plus thermocouple cold-junction compensation and linearization, engineering units conversion, auto-zero correction, auto-restart after power failure, real-time data display, spreadsheet and language-interfaces. No programming expertise is required on the part of the user.
- Labtech Notebook (PCI-20040S-1). This is a menu-driven software package for real-time data acquisition, process control, and/or data analysis. Labtech Notebook offers curve-fitting and Fast-Fourier-Transform (FFT) routines, on-line help and tutorials, graphic data display, and automatic interfacing to spreadsheet packages such as Lotus 1-2-3 and Symphony. It is excellent for users wanting a software package that requires minimum computer programming skills.
- Labtech Real Time Access (PCI-20065S-1). An extension to Labtech Notebook that supports real-time data transfers to spreadsheets or other analysis programs.
- DADiSP (PCI-20067S-1). A menu driven, graphical, data analysis package that is often called a scientific spreadsheet. It performs post acquisition signal analysis and display.
- SNAPSHOT STORAGE SCOPE (PCI-20068S-1). A menu driven waveform-capture system. Useful for transient analysis and general digital oscilloscope applications.
- RELAY LADDER LOGIC RS1000/PC (PCI-20073S-1). A ladder logic package for process control monitoring.

ADDITIONAL SOFTWARE

At the end of this section we list a number of software packages with the manufacturers' specifications. These packages are availalble directly from the manufacturers who have indicated to us that their product interfaces with the PCI-20000 system. These packages may be useful in data acquisition, test, and control applications. We can assume no responsibility for the suitability of these products and all warranties, representations, and further information should be obtained directly from the manufacturers, whose name and address we list on their specification pages.

PCI DEMONSTRATION DISKETTES

Several demonstration diskettes provide a clear and concise overview of PCI-20000 products. All diskettes run on IBM (and compatible) personal computers that contain a Color Graphics Adaptor (CGA) card. For your demonstration diskettes, please contact your local sales office listed at the back of this handbook.

- PCI-20000 System Demo (PCI-20034S-1). This diskette presents an overview showing system capabilities, specifications, and applications. Two levels of demonstration are provided: Basic Concepts and Advanced Features.
- PCI ControLOGraph Intelligent Data Logger Demo (PCI-20054S-1). This demonstration presents an overview of the PCI ControLOGraph showing how it is configured and what its capabilities are.
- Labtech Notebook Demo (PCI-20065S-1). This demo presents the major features of the menu-driven data acquisition and control package.
- SNAPSHOT STORAGE SCOPE DEMO (PCI-20069S-1). This diskette shows how SNAPSHOT emulates a digital storage oscilloscope. The demo diskette simulates real-product operations.
- DADiSP Demo (PCI-20072S-1). This demo shows the capabilities of the DADiSP analysis software.

BURR-BROWN®

PCI-20034S-1 PCI-20000 Demo
PCI-20054S-1 ControLOGraph Demo
PCI-20064S-1 LABTECH Notebook Demo
PCI-20069S-1 Snapshot Demo
PCI-20072S-1 DADiSP Demo

DEMONSTRATION SOFTWARE DISKETTES

FEATURES

- **Provides a clear and concise overview of PCI-20000 products**
- **'Lifelike' PC environment promotes understanding**
- **No PCI hardware is required**
- **Works with IBM PC, XT, AT and other compatibles**
- **Supplied on 5-1/4" diskettes with easy-to-follow instructions**

DESCRIPTION

The PCI-20034S-1 presents a general overview of the PCI-20000 System. System capabilities, specifications and applications are described. Two levels of treatment are provided: basic concepts and advanced features. Each level takes about 10 minutes to view. The user has complete control over presentation speed and can pause, go back or skip ahead at will. This material can be a valuable aid in system configuration as well as an appropriate source of background information for management and engineering.

The PCI-20054S-1 presents a general overview of the PCI ControLOGraph data acquisition and control system. The ControLOGraph is a menu driven software package specifically designed for data logging applications. Capabilities, specifications and applications are described. Two levels of treatment are provided: basic concepts and advanced features. Each takes about 10 to 15 minutes to view. The user can control the speed of presentation by pausing if desired. This material is valuable for both the potential end user and as an appropriate source of background information for management and engineering.

The PCI-20064S-1 provides a general overview of 'LABTECH Notebook'. LABTECH is a menu-driven software package for data acquisition and control. Capabilities and applications are suggested. Viewing takes about 10 minutes. The user can control the presentation rate. This material is valuable for both the potential end user and as an appropriate source of background information for management and engineering.

The PCI-20069S-1 provides a general overview of 'Snapshot Storage Scope'. Snapshot is a menu-driven data acquisition system that emulates a digital storage oscilloscope. Capabilities and applications are suggested. The user can simulate real product operations. This material is valuable for both the potential end user and as an appropriate source of background information for management and engineering.

The PCI-20072S-1 provides a general overview of 'DADiSP'. DADiSP is a menu driven, post acquisition, data analysis and display software package. It is a spreadsheet type product that operates on entire waveforms. Capabilities and applications are suggested. This material is valuable for both the potential end user and as an appropriate source of background information for management.

REQUIREMENTS

All disks require an IBM-compatible personal computer with a color graphics adapter (CGA). A color monitor is recommended; however, a monochrome monitor will suffice.

BURR-BROWN®

PCI-20046S SERIES*

PCI-20046S-1 BASIC Language Interface
PCI-20046S-2 C Language Interface
PCI-20046S-3 Turbo Pascal Language Interface
PCI-20046S-4 ASYST Language Interface
PCI-20046S-6 Combination Software Package containing PCI-20046S-1, PCI-20046S-2, and PCI-20046S-3

SOFTWARE DRIVERS
LANGUAGE SUPPORT SUBROUTINE LIBRARIES

FEATURES

- **Interfaces to BASIC, C, Turbo Pascal, ASYST and Assembly languages**
- **Compatible with PC/MS DOS, version 2.0 or higher**
- **Easy-to-use high-level commands**
- **Eliminates the need for the programmer to be familiar with the details of the hardware**

DESCRIPTION

The PCI-20046S series of software support packages are designed to provide an uncomplicated, consistent and useful interface between several of the most popular high-level languages and the PCI-20000 hardware system. These drivers maintain high performance while avoiding the complexities of Assembly language. All goals are realized, by buffering the programmer from the internal details, and by offering a set of commands to invoke the major hardware functions.

BASIC, C, Turbo Pascal and ASYST were selected because of their wide user base and general acceptance. However, all of the PCI-20046S packages can be used directly from Assembly language if desired. In fact, to optimize the speed of the function routines, the appropriate internal code has been written directly in Assembly language. Error checking is provided to help eliminate difficulties. Tutorial material, including detailed sample programs in each language, offers assistance to new users.

Please refer to the Speed Summary Table in the Configuration Section of this handbook for a comparison of language speeds.

*NOTE: The PCI-20046S series of software supersedes the original PCI-20014S series.

BASIC

IBM BASICA through version A3.10 is supported. Other true compatibles such as **Compaq BASIC** can be used. The interpreted version of **Quick BASIC** also works. BASIC has gained wide acceptance as a general-purpose language because it is easy to learn and utilize. Program development time is generally short. However, in comparison to other languages, its execution time is slow.

IBM BASIC Compiler through version 2.00 is supported. Other BASIC compilers are not recommended. Specifically, the Microsoft BASIC compiler is not suitable. Compiling BASIC code increases its execution speed about seven times, making it comparable to both C and Turbo Pascal.

C

Microsoft and Lattice C Compilers, through version 3.00 are supported. Microsoft version 4.00 using the Small, Medium and Large models are also supported. C is a versatile language that is respected by sophisticated programmers. Existing subroutines and libraries can be linked together to build a given application. Direct access to the computer is available. In contrast, however, C is considered cryptic and harder to learn than other languages. The execution speed of C is about seven times that of interpreted BASIC.

TURBO PASCAL

Borland Turbo Pascal through version 3.01A is supported. Turbo is considered very user friendly, with a host of built-in tools for I/O transactions. It is highly recommended for new programmers. Turbo offers all the power that a DA&C application requires, along with the speed of C, without much of the hassle. The relative execution speed of Turbo Pascal is about seven times that of interpreted BASIC.

ASYST

Macmillan ASYST through version 1.53 is supported. ASYST is a language (derived from FORTH) designed for scientific data acquisition applications. ASYST consists of several volumes. Volumes One and Two contain the actual language and general operating procedures. These products are available directly from the publisher, ASYST Software Technologies, 100 Corporate Woods, Rochester, NY 14623. Volume Three is replaced by the PCI-20046S-4, which contains the hardware interface software (drivers). This combination allows direct data acquisition from all PCI-20000 carriers and modules. The execution speed of ASYST is about three times that of interpreted BASIC (about half the speed of C or Turbo Pascal). Some of the unique features of ASYST include:

- Easy reduction, manipulation and analysis of acquired data
- Generation of high-quality graphics
- Quick comparisons of results using multiple-window displays
- Integration of analysis functions, including Fourier analysis, with graphics.

APPLICATIONS EXAMPLE

"Read an analog input channel."

BASIC

```
"
"        ' Include the Driver Interface Header
"        ' Initialize the Hardware & Software
"
120  CHN = 5         ' Define Channel 5
130  GAIN = 10       ' Define Gain = 10 (for Ch 5)
140  ZERO = 0        ' Autozero with respect to Ch 0
150  RANGE = 2       ' Define that A/D is on Bipolar range
"
"        ' Possible other tasks
"
230  CALL CNF.AI (CHN, GAIN, ZERO, RANGE)   'Configure Input
240  CALL READ.CH (AI, CHN, DATA)           'Read Analog Input
"
"
"        ' "Process / Manipulate / Display Data"
"          etc....
"          etc......
     END
```

C

```
cnf_ai (chn, gain, zero, range) ;
data = read_ch (ai, chn) ;
```

Turbo Pascal

```
Procedure CNfAI (Chn, Gain, Zero, Range);
Data = Read Ch (AI, Chn);
```

COMMAND SUMMARY—PCI20046S SERIES

Utility Calls

AUTOGRAPH	Verify compatibility of installed software
ERROR.SYS	Return the last error code
INIT	Initialize the system hardware
SYSINIT	Initialize the system software

Configuration Calls

CNF.AI	Configure an analog input channel
CNF.CNTR	Configure a counter channel
CNF.DI	Configure a digital input
CNF.DO	Configure a digital output
CNF.RG	Configure a rate generator
CNF.RTD	Configure an analog input as an RTD input
CNF.TCPL	Configure an analog input as a thermocouple input
CNF.TRIG	Configure a trigger alarm module to start acquisition

Read Calls

READ.CH	Read any input channel
READ.CTS	Read a group of three counters simultaneously
READ.FRQ	Read the frequency of a TTL input
READ.SSH	Read simultaneous sample/hold inputs
STAT.CNT	Read a counter value and status

Write Calls

WRITE.GR	Write a group of analog outputs simultaneously
WRITE.CH	Write data to any output channel

Miscellaneous Calls

CVT.RTD	Linearize RTD data
CVT.TCPL	Linearize and compensate thermocouple data

ASYST

```
CALL [ PCI20K, CON.AI, CHN, GAIN ]
CALL [ PCI20K, READ.CH, AI, CHN, DATA ]
```

BURR-BROWN®

PCI-20047S-1

High-Performance, High-Speed Software Driver Extension Library

FEATURES

- **Supports Data Files Up to the Limit of Available 640k RAM**
- **Supports Direct Memory Access (DMA)**
  - **-Up to 360k bytes/s I/O transfers**
  - **-Analog, digital, and counter data can all be transferred simultaneously**
- **Supports High-Speed Analog Data Acquisition**
  - **-Up to 89kHz throughput**
  - **-Up to 178kHz with DMA**
- **Provides for Pre-Trigger and Post-Trigger Viewing of Data**
- **Supports all PCI-20000 System Carriers and Instrument Modules**
- **Has Easy-to-Use High-Level Commands**
  - **-Programmer needs only limited familiarity with hardware details**
- **Offers Multiple Language Interface**
  - **-Used with PCI-20046S series of software support drivers for: BASIC, C, Turbo Pascal™, ASYST™**

DESCRIPTION

The PCI-20047S-1 high-performance software diskette operates in IBM PC, XT, AT, and compatible personal computers and provides support for the PCI-20000 data acquisition, test, measurement and control system.

The PCI-20047S-1 is a collection of assembly language routines which can be called from a high-level language and thus isolate the programmer from having to interface directly with hardware. The PCI-20047S-1 is used in conjunction with the PCI-20046S series to provide interface calls to selected programming languages—BASIC, C, Turbo Pascal, or ASYST. The two software packages are thus "linked" together to provide direct memory access (DMA) calls and high-speed data acquisition calls to PCI-20000 hardware.

The DMA calls offered by the PCI-20047S-1 allow analog, digital, and counter data transfers to occur simultaneously and support the PCI-20041C-3 high-performance carrier in the functions of foreground/-background mode, multiple I/O function capability, and fast data transfers. The foreground/background feature allows the user to start a DMA process, and then while the DMA is operating, to perform whatever other functions are desired in the user's programming language. For example, the DMA can be used to collect analog input data and to store it in memory in the background. At the same time the foreground program (under user software program control) can be performing display, analysis, or control functions. DMA transfers allow data files greater than 64k bytes, up to the limit of available RAM, within the lower 640k bytes. Data files can be written to and read from simultaneously. Multiple buffers can be allocated.

The PCI-20047S-1 allows the maximum speeds possible with the PCI-20000 system. This speed is 360,000 bytes/s for digital I/O transfers in the IBM PC, and 168,000 bytes/s in an IBM AT computer. Please note that the speed of the DMA in the IBM PC is faster than the speed in the IBM AT because of the IBM AT's internal architecture. Please refer to the Speed Summary Table in the configuration section of this handbook, for obtainable speeds using DMA.

The high-speed data acquisition calls that the PCI-20047S-1 supports will work with all carriers and with all instrument modules in the PCI-20000 system. They allow a user to make a block-mode reading, and to specify the channels to read, the sampling frequency, and the number of samples to use in acquiring data. The sampling frequency can be

provided by a pacer clock on a PCI-20041S series carrier, by a rate generator on a PCI-20007M-1 counter/timer instrument module, or it can come from a separate external source. Throughput rates are shown in the Speed Summary Table.

The PCI-20047S-1 provides for other extremely important data acquisition, test, measurement, and control functions. The multiple I/O function capability allows a user to input a variety of input types all at the same time, or output a variety of output types at the same time. For instance, the user can read analog inputs, digital inputs, and counter inputs in the same DMA setup. In the same manner, the user can write analog outputs and digital outputs in the same DMA setup. Often it is highly desirable to examine the data acquired just before and just after a specified trigger event. The PCI-20047S-1 provides for this important pre-trigger/post-trigger function also.

Specific commands are shown in the Command Summary Table.

PCI-20047S-1 COMMAND SUMMARY

Configuration Calls	
CNF.DMA	Configure a DMA data acquisition.
CNF.HS	Configure a high speed data acquisition.
Data Acquisition Calls	
DMA.RUN	Start a DMA data acquisition.
DMA.STOP	Stop a DMA data acquisition.
HS.RUN	Execute a high speed data acquisition.
Data Transfer Calls	
DMA.READ	Transfer DMA data to input buffer.
DMA.WRITE	Transfer DMA data from output buffer.

BURR-BROWN®

PCI-20040S-1

LABTECH® NOTEBOOK
Integrated Data Acquisition, Control and Analysis Software for Personal Computers

FEATURES

- Compatible with most PCI-20000 components (see Specifications)
- Compatible with PCI-3002/3003 (PCI-3307-1 and PCI-3307-2)
- Operates with all IBM PCs and compatibles (PC, XT, AT, Compaq, AT&T, Zenith, etc.)
- Easy-to-use, menu-driven software
- Real-time data acquisition
- Real-time process control—open or closed loop
- Real-time analysis and data reduction
- Real-time graphics display of data
- Foreground/background operation under MS/PC DOS
- Built-in curve-fitting and FFT routines
- Automatically interfaces with related products such as: Real Time Access, Labtech Chrom and Lotus 1-2-3

DESCRIPTION

PCI-20040S-1 is the model number for the industry standard LABTECH Notebook, designed for PCI products (see Figure 1). Notebook couples to the major hardware elements within the PCI system. Data acquisition, control and analysis tasks are reduced to menu-driven choices—minimum computer skills are required. Programming options allow the automation and customization of advanced analysis, presentation, and report generation features. Voltage, current, thermocouple, RTD, pulse, frequency and digital input data can be recorded and displayed in real-time while open- and closed-loop proportional-integral-derivation (PID or on/off) outputs are generated. In addition to digital outputs, both voltage and current analog outputs are available.

LABTECH Notebook is an integrated, general-purpose software package for data acquisition, monitoring and real-time control. It runs on the IBM PC, XT, AT and other PC compatible computers. Notebook operates with the PCI analog/digital interfaces, and insulates the user from the low-level instructions the hardware often requires. It replaces laboratory notebooks and hand-keying of data in the same way that spreadsheet programs such as Lotus 1-2-3 replaced paper spreadsheets in business offices.

Because LABTECH Notebook is menu-driven, and extemely easy to learn and use, it requires very little computer skill on the part of the operator. The conditions which define the current run are displayed on the screen and are readily modified. All of the conditions pertaining to a run can be easily saved or recalled as a group. LABTECH Notebook reduces complicated data acquisition and control procedures to single-button operations, so that repetitive tests and process-monitoring activities are greatly simplified.

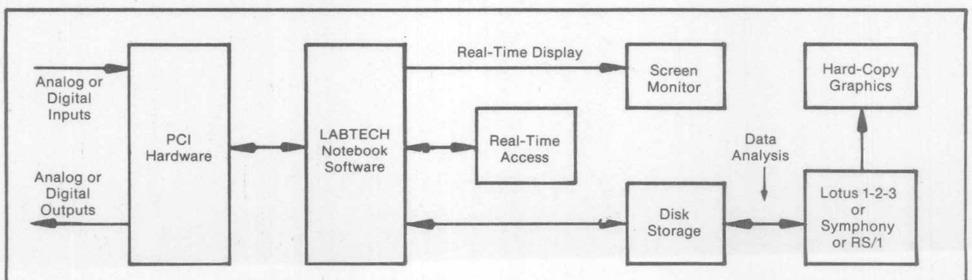


FIGURE 1. LABTECH Notebook Functional Diagram.

There is a programming option available for those who wish to customize or automate complicated data acquisition or process control procedures. This option uses a nonextensible version of MAGIC/L™, an incrementally compiled language.

Flexibility is a key feature of LABTECH Notebook. To this end, each channel can be set up with different characteristics. Sampling rates may vary from channel to channel, and on each channel the sampling rate may vary at different times during a run. Notebook's data monitoring and filing capabilities are equally versatile.

Channels can be used for purposes other than simple inputs or outputs. LABTECH Notebook has the ability to derive channels from other channels. For example, channels can 'operate' on others by calculating averages, derivatives, integrals, etc., in real time. The list of mathematical, logical, statistical and signal processing functions is quite extensive. These derived channels can also be used in determining triggers or as inputs to control loops.

Stored data can be "played back" as though it were being acquired in real-time. This provides a facility for making comparisons with data previously acquired or with data that has been theoretically derived.

Open- and closed-loop control algorithms are readily implemented. In open-loop mode, the user defines one period of any imaginable waveform and the signal is then clocked out automatically during the run. For closed-loop control, both proportional-integral-derivative (PID) and 'bang-bang' (on/off) loops can be set up.

LABTECH Notebook includes a powerful curve-fitting function. It uses an iterative routine to fit an arbitrarily complex model (up to ten parameters) to the collected data. This routine can be set up to take advantage of the PC's optional 8087 (or 80287) co-processor. The 8087 offers 80-bit real number processing, reduces round-off error, and allows faster computation.

When the user purchases an analysis or spreadsheet program, such as DADiSP, Lotus 1-2-3, or Symphony, additional functionality is gained. LABTECH's data files can be automatically imported into one of these programs for further reduction or analysis. Some of the programs will also provide word processing or data base capabilities. It is possible to build a 'seamless' integration between LABTECH and these other programs, making it easy to go from acquisition to a final report without writing down or keying in a single piece of data.

BASIC OPERATION

LABTECH Notebook is a menu-driven program. There are no commands for the user to remember. Setup conditions for data acquisition, process control, data storage, and real-time display are all entered into 'option' tables.

An option table displays a list of setup conditions on the left-hand side of the screen. The corresponding values for each of the setup conditions appear in the column to the right of the list. This is the basic format of all of Notebook's option tables. Changing any value is easy. Simply move the 'cursor' (highlighted rectangle) up or down the entry column (using the cursor control keys) until the appropriate value is located. A new value is simply typed in and the Enter key is pressed.

DATA ACQUISITION

LABTECH Notebook can perform data acquisition in either the normal or a high-speed mode. In the normal mode, acquisitions may be performed at sampling rates from 0.001Hz to 900Hz. In this mode, real-time display of data is available to the user, and data may also be permanently stored in user-defined files. In the high-speed mode, rates up to 25kHz can be obtained. However, the maximum speeds depend upon the particular PCI and computer hardware being used.

In the normal mode each channel may have different setup conditions. That is, they may have different channel types, scale factors, sampling rates, etc. As Figure 2 illustrates, the time period for each channel may be divided into several stages, each stage having a different sampling rate, duration, and/or starting method.

The series of stages may be repeated by setting an appropriate iteration count for the channel. Figure 3 shows a channel which has been set up to go through several loops, each consisting of a pass through all the channel's stages (except the first, which is used only during the first loop).

Any stage may be initiated in one of three ways:

- Normal starting, where a stage begins as soon as the previous stage ends (the first stage begins as soon as the run is initiated).
- Trigger starting, where a digital input or an analog level on any channel (or combination thereof) is received before the stage can begin.
- Time delay starting, where a stage begins only after a user-specified time has elapsed.

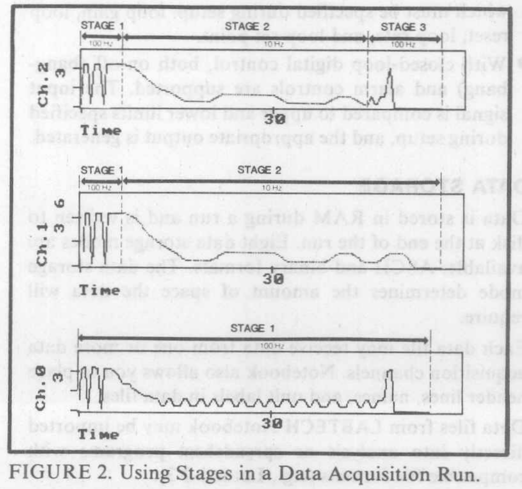


FIGURE 2. Using Stages in a Data Acquisition Run.

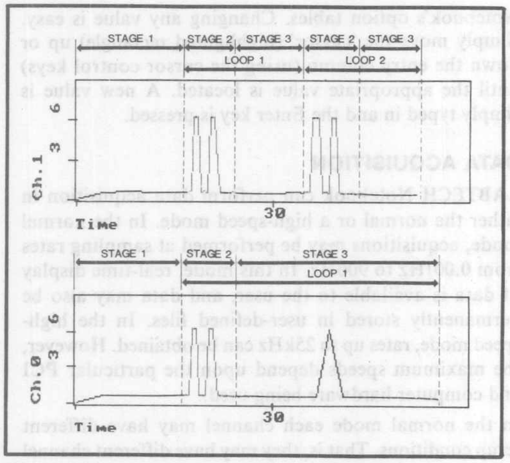


FIGURE 3. Using Loops in a Data Acquisition Run.

A stage may run until:

- A stated time period has elapsed, or
- An analog or digital trigger occurs.

In the high-speed mode, real-time display of data is not available. All channels have identical setup conditions, and multiple stages and loops are not available. A high-speed run may be started using either the normal, trigger, or time-delay method.

PROCESS CONTROL

Both open- and closed-loop process control are available using LABTECH Notebook.

- When open-loop control is specified, the contents of a data file are sent, point-by-point, to the hardware interface.
- With closed-loop analog control, the output is determined according to a PID equation. This equation provides an output signal which is a function of the input from an A/D channel and four PID variables which must be specified during setup: loop gain, loop reset, loop rate, and loop set point.
- With closed-loop digital control, both on-off (bang-bang) and alarm controls are supported. The input signal is compared to upper and lower limits specified during setup, and the appropriate output is generated.

DATA STORAGE

Data is stored in RAM during a run and is written to disk at the end of the run. Eight data storage modes are available: ASCII and binary formats. The data storage mode determines the amount of space the data will require.

Each data file may receive data from one or more data acquisition channels. Notebook also allows you to place header lines, names, and unit labels in data files.

Data files from LABTECH Notebook may be imported directly into analysis or spreadsheet programs with compatible file formats (e.g., Lotus 1-2-3).

REAL-TIME DISPLAY

The real-time display function is available during normal mode data acquisition and control runs. Figure 4 shows an example of the display. As can be seen, the display is in the form of X-Y graphs, Y-time graphs, vertical bars, digital meters, and (not shown) horizontal bars. Up to 50 signals can be displayed in up to 15 windows. The presentation and scaling of the displays are under the control of the user. Scrolling of the horizontal axis is provided in the Y-time mode.

LABTECH supports several display standards, including CGA (4 colors), EGA (8 colors), Hercules and Compaq monochrome.

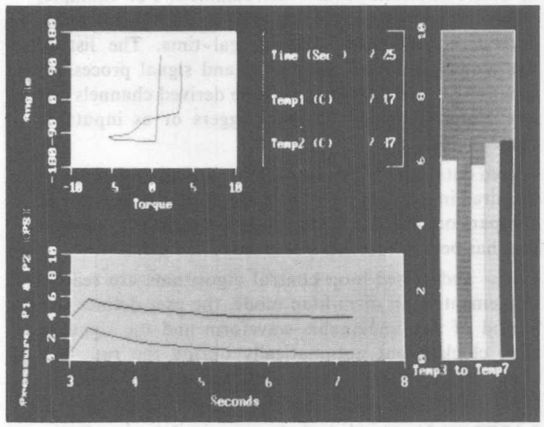


FIGURE 4. Real-Time Display.

DATA ANALYSIS

LABTECH Notebook has been designed to support both real-time and post process analysis. For post process work, files can be interfaced to other products such as DADiSP and Lotus 1-2-3 as well as user-written programs. Since LABTECH files can be generated in several different formats, the one most efficient for a particular purpose can be selected.

For example, data manipulation, statistics, and data base management are all available using Lotus 1-2-3. In addition, publication-quality graphs may be generated using 1-2-3's PrintGraph function.

An analysis or spreadsheet program can be invoked directly from the Notebooks' main menu. There is no need to leave Notebook.

Using spreadsheet programs can greatly simplify post-processing tasks. Lotus 1-2-3, for example, can set up and save worksheet templates which reduce data manipulation and graphing functions to single-key operations.

NONLINEAR REGRESSION ANALYSIS—CURVE-FIT

Notebook offers a curve-fitting routine which enables you to fit an arbitrarily complex model to experimental data.

A mathematical model consisting of independent variables and parameters is entered by the user. Initial estimates for the parameters are also entered. Along with this equation, a data file with experimental data is provided.

The routine uses the model to calculate theoretical values for the data using the initial parameter estimates. The difference between experimental and theoretical data, the residual, is then calculated, and the squares of the residuals for every data point are summed. New values for the parameters are then chosen by the routine in an attempt to minimize this sum of the squared residuals (SSQR).

With the new choices for the parameters, theoretical values are again calculated, and a new set of residuals is found. Given the last value of SSQR and the new value, the routine can determine in which direction to vary the parameters so the SSQR will grow smaller. The process is repeated until the change in parameters between iterations is less than the specified parameter tolerance.

Full statistical data is presented on the quality of the fit, the accuracy of the fitted parameters, and inter-parameter correlation. A graph showing both the experimental and theoretical curves can be produced. Figure 5 shows an example of such a graph. An analysis of variance is also performed by the curve-fitting function. The output from this analysis consists of estimates of the parameter standard deviations and a parameter correlation matrix. These results allow the user to determine the quality of the fit to the theoretical model.

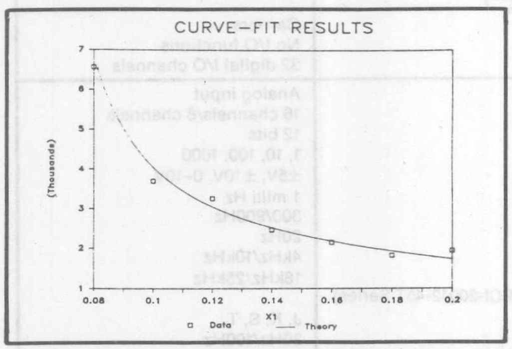


FIGURE 5. Graph of Theoretical Data Vs Experimental Data—Output from Nonlinear Regression Analysis.

PROGRAMMING OPTION

For those who want to automate data acquisition or process control runs, Notebook offers a programming option. Using this option the user can write routines to start and repeat runs, perform filing chores between runs, and branch on conditions.

The commands available under this option are a subset of the MAGIC/L programming language. They fall into the following categories:

- Branching and control
- Input/output
- Looping
- Notebook-defined functions (to start runs, etc.)
- DOS commands (can be used to execute DOS programs)
- Arithmetic functions.

The programming option is selected from the main menu. Since MAGIC/L is compiled line-by-line, routines do not have to be compiled separately; they are typed in and executed immediately. Stored programs written in the MAGIC/L language may also be executed using this option.

Among the programming possibilities offering by Notebook is an auto-start file which contains a MAGIC/L program; when the program is started, the file is executed, allowing data acquisition or process control procedures to be initiated automatically.

PRODUCTION LINE TESTING

Once the setup conditions and data analysis worksheets have been developed, LABTECH Notebook can be configured for production line testing very easily. All that is required is a very short program written in Notebook's programming option language, MAGIC/L. This program can either be entered at the console and executed immediately, or stored in an autostart file which will be executed when the computer's power is turned on.

A program to collect an entire shift's worth of data, say 100 pressure vessel tests, might look like the following:

```
iter 100    ;Repeat routine 100 times
     dogo   ;Invoke GO function
     r_123  ;Invoke 1-2-3, perform analysis
loop        ;Return to first command
```

This simple program will perform 100 data acquisition runs. Each data acquisition will be automatically triggered by an electrical signal from the production line. After each run, the data files for the run will be immediately analyzed. The results from all the runs will be accumulated in a data base file (using 1-2-3's macro capability), thus completely automating the shift's testing.

DOS SYSTEM UTILITY

If at any time during a LABTECH Notebook session the user wishes to run a program under the DOS operating system, the DOS system utility may be selected from the menu. Scientific application programs, word processing programs, or specialized user-written programs may be executed using this function.

SPECIFICATIONS—PCI-20040S-1

When a reference is made to a PCI-20000 part without a "-#" (dash number), it applies to all available dash numbers.

PARAMETER	CONDITIONS	SPECIFICATION
LABTECH Notebook		Version 4.0
Compatibility	PCI-20001C Carriers, PCI-20002M, PCI-20003M, PCI-20004M,	
	PCI-20005M, PCI-20007M, PCI-20019M, PCI-200021M Modules	
	and most PCI-20000 Termination Panels and Accessories	
	PCI-3002/3 Master Enclosures and PCI-3307 Boards	
Data Acquisition		
Sampling Rate	See Hardware items below	up to 89kHz
Run Duration		up to 270k hours
Input Types		Analog, digital, pulse, frequency
Starting Methods		Delayed, triggered
Process Control		
Sampling Rate	See Hardware items below	up to 250Hz
Control Modes		Open loop and PID
Input Types		Analog, digital, pulse, frequency
Output Types		Analog and digital
Data Storage		
Types		RAM and disk
Modes	ASCII real and integer, binary real and integer; hex; binary	Real and integer
Real-Time Display		
Speed	Maximum[4]	900Hz
Types		Y-time, X-Y, bars, meters
Number	Screen partitions	up to 15
Colors		Four (CGA), eight (EGA)
PC Requirements		
Type	IBM PC, XT, AT or fully compatible	
Drives	Hard disk recommended, two diskettes functional	
Memory	384K RAM, 512K RAM recommended	
DOS	2.0 or greater, 3.0 or greater recommended	
Display	Graphics board required for graphics display	
PCI-20001C		Carriers
Channels[1]	PCI-20001C-1	No I/O functions
	PCI-20001C-2, see PCI-20004M below	32 digital I/O channels
PCI-20002M-1 [2]	Can be used with PCI-20005M Expander	Analog input
Channels[1]	Single-ended/differential	16 channels/8 channels
Resolution		12 bits
Gain	Programmable	1, 10, 100, 1000
A/D Ranges	Jumper selected	±5V, ±10V, 0-10V
Sampling Rate	Normal mode, minimum	1 milli Hz
	Maximum, PC/AT	300/900Hz
	High speed, minimum	20Hz
	Maximum, PC/AT	4kHz/10kHz
	Free Run, approximate, PC/AT	16kHz/25kHz
Thermocouples	Using PCI-20010T-2 or PCI-20057T-1 (Not PCI-20042-45T Series)	
Types		J, K, S, T
Sampling Rate	Maximum, PC/AT	30Hz/100Hz
PCI-20003M		Analog output
Channels[1]	PCI-20003M-1, voltage output	One channel
	PCI-20003M-2, voltage output	Two channels
	PCI-20003M-3, current or voltage output	One channel
	PCI-20003M-4, current or voltage output	Two channels
Resolution		12 bits
D/A Ranges	Voltage output	±5V, ±10V, 0-10V
	Current Output	4-20 and 5-25mA
Data Rate	Open loop, maximumm, PC/AT	300Hz/1400Hz
	PID, Maximum, PC/AT	20Hz/70Hz[3]

SPECIFICATIONS (CONT)

PARAMETER	CONDITIONS	SPECIFICATION
PCI-20004M-1	Specs also apply to PCI-20001C-2	Digital input/output
Bytes[1]	Eight channels each, programmable as I/O	Four
Logic Levels		TTL
Sampling Rate		
Input	Maximum, PC/AT	300Hz/900Hz
Output	Maximum, open loop, PC/AT	400Hz/1500Hz
	Maximum, closed loop, PC/AT	70Hz/250Hz
PCI-20005M-1[2]	Can be used with PCI-20002M	Analog expander
Channels[1]	Single-ended/differential in normal mode only	32 channels/16 channels
PCI-20007M-1		Counter/timer
Counter Channels[1]	Count and measure frequency (±1Hz)	Four channels
Count	Maximum	65,535 channels
Read Rate	Maximum, PC/AT	300Hz/900Hz
Input Speed	Maximum input frequency	8MHz
Pulse Output	Frequency range (duty cycle dependent)	Up to 8MHz
PCI-20019M-1	High speed	Analog input
Channels[1]	Single-ended	Eight channels
Resolution		12 bits
A/D Ranges	Jumper selected	±5V, ±10V, 0-10V
Sampling Rate	Normal mode, minimum	1 milli Hz
	Maximum, PC/AT	300Hz/900Hz
	High speed mode, minimum	20Hz
	Maximum, PC/AT (Requires PCI-20007M-1)	50kHz/80kHz
PCI-20021M-1		Analog output
Channels[1]	Voltage output	8 channels
Resolution		12 bits
D/A Ranges		±5V, ±10V
Data Rate	Open loop, maximum, PC/AT	300Hz/900Hz
	PID, maximum, PC/AT	20Hz/70Hz[3]
PCI-3002	Specs apply to PCI-3307-1	Combination I/O
Analog Input		
Channels[1]	Single-ended/differential	16/8
Resolution		12 bits
Gains	Programmable	1, 201[5]
A/D Ranges		±2.5V, ±5V, ±10V, 0-5, 0-10V
Sampling Rate	Normal mode, minimum	1 milli Hz
	Maximum, PC/AT	16Hz
	High speed, maximum, PC/AT	1kHz[6]
	Free run, approximate, PC/AT	2kHz
Thermocouples		
Channels[1]		8
Types		J, K, T
Sampling Rate	Maximum, PC/AT	16Hz
Analog Output	PCI-3003 and PCI-3307-2 only	
Channels[1]	Voltage output	2
Resolution		12 bits
D/A Ranges		±2.5V, ±5V, ±10V, 0-5, 0-10V
Data Rate	Open loop, maximum, PC/AT	16Hz
	PID, Maximum, PC/AT	16Hz
Digital Input/Output		
Bytes[1]	8 channels each, programmable as I/O	2
Logic Levels		TTL
Sampling Rate		
Input	Maximum, PC/AT	16Hz
Output	Maximum, open loop, PC/AT	16Hz
	Maximum, closed loop, PC/AT	16Hz

NOTES: (1) The number of channels apply to "each" module used. (2) LABTECH supports a maximum of two PCI-20005M-1 Expanders with each PCI-20002M-1 A/D Module. (3) Using an 8087/80287 coprocessor greatly increases speed. (4) Speed without additional supplementary hardware (i.e., 'Scroller Card'). (5) Other gains can be resistor selected by the user. (6) To internal 32K RAM.

BURR-BROWN®

PCI-20056K-1

Complete System

PCI ControLOGraph™
A Fully Integrated Data-Logging System for IBM-Compatible Personal Computers

FEATURES

- Integrated data-logging, real-time display, alarm, control and graphic analysis package
- Data recorded in hours, minutes, seconds, and tenth seconds along with day, month, and year
- Completely menu driven, NO programming required, NO previous computer skills required
- 48 inputs—21 differential analog inputs, including:
  - -thermocouples
  - -24 digital inputs
  - -3 counter/frequency inputs
- 8 digital alarm and control outputs controlled by analog, digital, and/or counter channels
- Trigger options (analog, digital, and/or counter) allow both pre- and post-trigger data to be viewed
- Auto-restart allows an unattended system to resume taking data after recovery from a power failure
- Graphics analysis incorporates "active cursors" to display a given data point or the difference between data points
- ASCII data files are compatible with Lotus, ASYST, BASIC, etc., providing for additional post-acquisition analysis
- System includes PC plug-in carrier board, modules, termination panels, cables, software, and operations manual

APPLICATIONS

- General data-logging functions, replaces chart recorders and printers while providing extensive analysis capabilities
- Time studies
- Laboratory data collection and control
- Life test and burn-in operations
- Utilities monitoring

See PCI-20056K-1 in the Hardware Section for complete information.

BURR-BROWN®

PCI-20065S-1

LABTECH REAL TIME ACCESS
Software for Real Time Data Analysis and Control

FEATURES

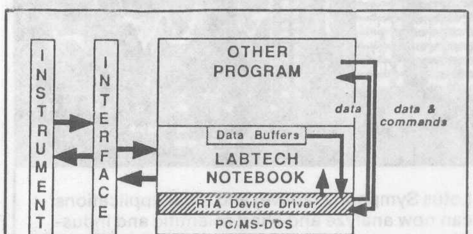
- Operates in LABTECH NOTEBOOK's foreground/background environment
- Works as an interprogram communications link between LABTECH NOTEBOOK and most application programs
- Can be used with popular spreadsheets, data bases, and analysis programs—such as 1-2-3, Release 2, Symphony, and dBASE III
- Can work with vertical applications like LABTECH CHROM
- Compatible with most programming languages
- Access is through an installable device driver
- Programs access real time data through standard file I/O statements
- Provides for real time acquisition of data into the application program
- The application program can write data to instrumentation
- Allows the program or console operator complete control over the data acquisition environment.
- Provides for real time modification of NOTEBOOK's process control environment
- Provisions made for the synchronization of the application program to the real time environment

DESCRIPTION

LABTECH NOTEBOOK has for some time featured the unusual capability of operating as a real time multitasking subsystem on a personal computer with PC/MS-DOS. NOTEBOOK can be 'in the background' carrying out a complex schedule of data acquisition and control while the user at the console is running some other DOS application program. Until the advent of Real Time Access, the two programs ran independently, and did not communicate with each other.

Real Time Access provides for that interprogram communication. It allows the foreground application to access the real time data being acquired simultaneously by NOTEBOOK. To the application program, that data looks as if it is coming from an ordinary DOS file or device. The user gains great flexibility in managing real time data analysis and control.

This gain in flexibility and power comes whether or not you write your own programs, since virtually any existing application program that can access files can access data acquired in real time from analog or digital instrumentation by LABTECH NOTEBOOK. It need not have been specially written to work with the LABTECH RTA device, since the device is indistinguishable from an ordinary file from the foreground application program's point of view. So existing spreadsheets, statistical analysis programs, quality control programs, graphics programs and more can gain real-time data acquisition capabilities.



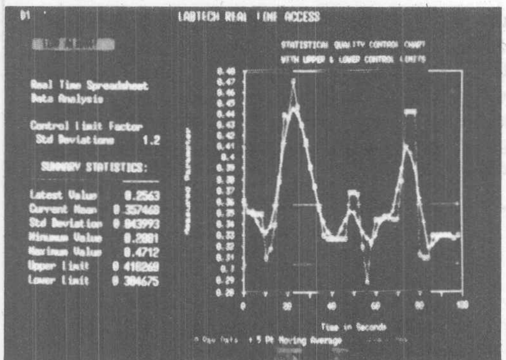
NOTEBOOK acquires data from instrumentation and places it into a data buffer area in memory. This buffer may be freely accessed by the application program, which can input real-time data by using ordinary file reading operations.

Foreground programs communicate with NOTEBOOK running in the background. They may access any or all of the data being acquired by NOTEBOOK data acquisition hardware drivers under control of the NOTEBOOK scheduler, or the foreground programs may initiate data acquisition or process control events by themselves.

Access is via a DOS device driver, communicating with a "porthole" in LABTECH NOTEBOOK's code. The driver creates a virtual real time data acquisition and process control device appearing to the application program as an ordinary device or file. Therefore, the program can communicate to NOTEBOOK by sending commands using standard write statements in the language of the foreground program.

Reading is done in one of several modes. In one mode, all data accumulated in the buffer since the last read request is passed to the foreground program, assuring that no data will be skipped. In another mode, only the latest data point in the buffer is passed to the foreground program. Yet another command allows for the immediate collection of a new data point regardless of when the last one had been collected and put in the buffer. Reading can also be set to return with no data if none has been collected since the last read command, or to wait for data to be entered in the buffer. This last feature provides a simple way for the foreground program to synchronize itself to the data acquisition rate.

The application program may optionally write a command stream modifying NOTEBOOK's acquisition and control parameters, or sending data directly to instrumentation. These commands may be sent programmatically, or by simply copying a file or direct console input to the device driver.



Lotus Symphony (above) and other applications can now analyze and chart scientific and industrial process data in real time with NOTEBOOK Real Time Acess.

Following is a command list and a brief explanation of their function.

Command	Function
Data Mode	Sets the mode of taking data from the buffer
Channel Select	Selects the channels to be read by the program
Set Channel On/Off	Turns data acquisition and control on and off
Data Format	Defines the format of the data strings
Termination String	Defines the end-of-line terminator
End-of-File Mode	Defines when an EOF condition is sent
Hold Mode	Toggles between read with wait and read with immediate return
Read Immediate	Gets data from the instrumentation, bypassing the buffer
Write Immediate	Sends data directly to the instrumentation
Set Setpoint	Modifies the PID setpoint
Set Gain Constant	Modifies the PID gain
Set Reset Constant	Modifies the PID reset constant
Set Rate Constant	Modifies the PID rate constant
Set Upper Limit	Sets the upper limit for digital control
Set Lower Limit	Sets the lower limit for digital control
Restore	Restores all defaults

Hardware Requirements

- IBM PC, PC XT, PC AT, or 100% compatible
- 384kbytes of memory plus memory required for user or analysis software
- Two double sided, double density 5-1/4 diskette drives (hard disk recommended)

Software Requirements

- PC-DOS revision 3.1 or higher
- LABTECH NOTEBOOK revision 3.0 or higher

PCI-20000 Hardware Supported

See LABTECH NOTEBOOK Data Sheet (PCI-20040S-1) for additional information.

BURR-BROWN®

PCI-20067S-1

DADiSP WORKSHEET

Scientific Spreadsheet for Post-Acquisition Signal Analysis

FEATURES

ANALYSIS CAPABILITIES
- Waveform Generation
- Peak Analysis
- Signal Editing
- Trigonometric Functions
- Calculus Functions
- Fourier Analysis
- Frequency Domain Analysis
- Time Series Analysis
- Statistical Analysis
- Complex Number Support
- Engineering Units Processing

GRAPHICS CAPABILITIES
- One to 64 Analysis Windows
- Scroll
- Zoom
- Cursor
- Table Display

PROGRAMMING CAPABILITIES
- Virtual Signal Management
- Macro Definitions
- Command Files

DESCRIPTION

The DADiSP Worksheet provides menu-driven, post-acquisition data analysis and display. It does for science and engineering what financial spreadsheets have done for business. It eliminates the need to spend valuable time programming specialized routines. DADiSP allows the user to set up a worksheet, or analysis chain, simply by typing formulas.

Each worksheet contains windows analogous to cells in a financial spreadsheet—except that a DADiSP window manages an entire waveform. The user can define up to 64 of these windows in a given worksheet. Every window acts as a processing step and displays a graph of the data transformed by that step. Any change in the data in one window is automatically updated in all the other windows using that data.

Because DADiSP pages the signal to disk as it is scrolled through a window, the software can handle signals of virtually any length. The program processes as much signal as the memory can handle, stores the processed data, calls up the next batch of signal data from the disk and continues until all of the signal data is processed. The user can zoom any window to full screen size for detailed manipulation and study.

Pop-up boxes provide background information on a window and carry scientific units through compound calculations. The Worksheet data is stored in individual DADiSP Laboratory books that permit individual users of the DADiSP system to keep their work separate from other users.

To analyze or manipulate the data, the user simply types in the desired command or formula into the desired window using any of more than 150 built-in functions. The functions include 42 trigonometric, calculus and common arithmetic operations; 14 Fourier analysis and related functions such as fast Fourier transforms and complex algebra; ten statistical functions, 21 signal editing and peak analysis operations, and 32 functions for generating complex waveforms.

Special user-defined functions are easily generated by calling up a user macro table and typing in the desired formula for the macro. For example, an autocorrelation macro could be generated by using a number of the standard DADiSP functions such as "convolute two signals." And a chain of operations can be set up by typing in the formulas in consecutive windows—with the results of each step displayed in sequence, window by window.

The DSP Pipeline function allows the running of external programs, such as Labtech Notebook or the user's own analysis routines, within the DADiSP environment and the importing of new or modified data directly into a worksheet.

For example, an engineer with a data acquisition program on an IBM AT can write a simple Pipeline command file in DADiSP to:

1) run the acquisition
2) import experimental data directly into a DADiSP analysis worksheet.

Using the DSP Pipeline, a scientist can, with a single keystroke, export data from any DADiSP window (i.e., data processing step) to an external set of analysis or filtering algorithms and bring the modified data right back into DADiSP.

Pipeline adds flexibility to DADiSP by dynamically linking the Worksheet environment to user-written or commercially available software.

DADiSP supports a variety of printers for hardcopy output and can even send data files to external plotter drivers to customize graphs.

DADiSP can import and export data files in a variety of formats including ASCII, Lotus PRN, byte, integer and floating point.

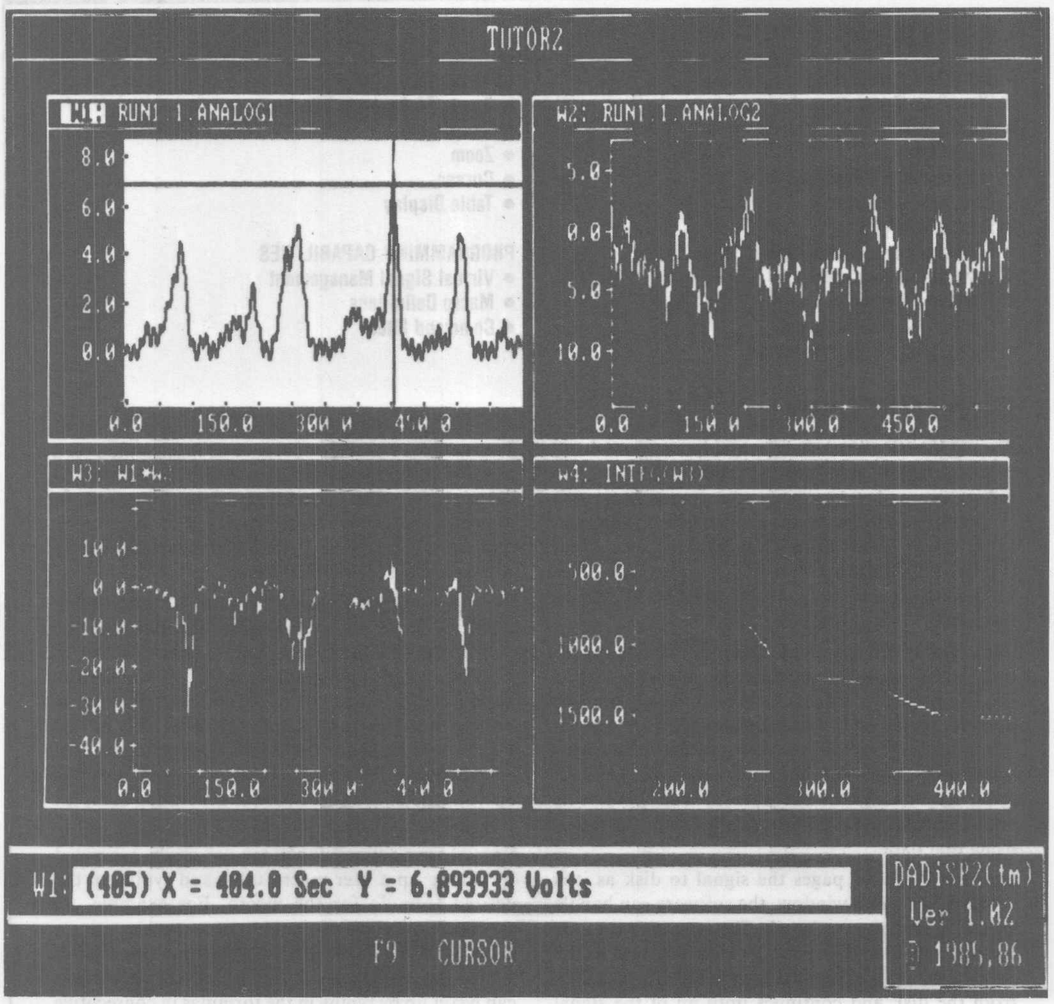


FIGURE 1. Cursor Function Allows the User to Scroll Through Signals Point by Point to Analyze Data Values at Critical Points.

HARDWARE REQUIREMENTS

- IBM PC, XT, AT or compatibles
- 512k bytes memory (640k bytes recommended)
- Optional 8087 or 80287 Co-Processor (recommended)
- One 360k bytes floppy drive with second floppy or hard drive

GRAPHICS REQUIREMENTS

- IBM Color Graphics Adapter, or
- IBM Enhanced Graphics Adapter, or
- Hercules Monochrome Adapter

SOFTWARE REQUIREMENTS

- IBM PC-DOS™ 2.0 or higher

PRINTER SUPPORT

- Epson FX/MX 80
- Okidata
- Gemini
- HP Laserjet
- HP Thinkjet
- Corona Laser Printer
- IDS Prism
- Apple Imagewriter
- Diablo Ink Jet Color Printer
- ACT II Printer

DATA FILE SUPPORT

- ASCII data
- Lotus PRN files
- 8-bit byte data
- 16-bit integer data
- IEEE 32-bit floating point

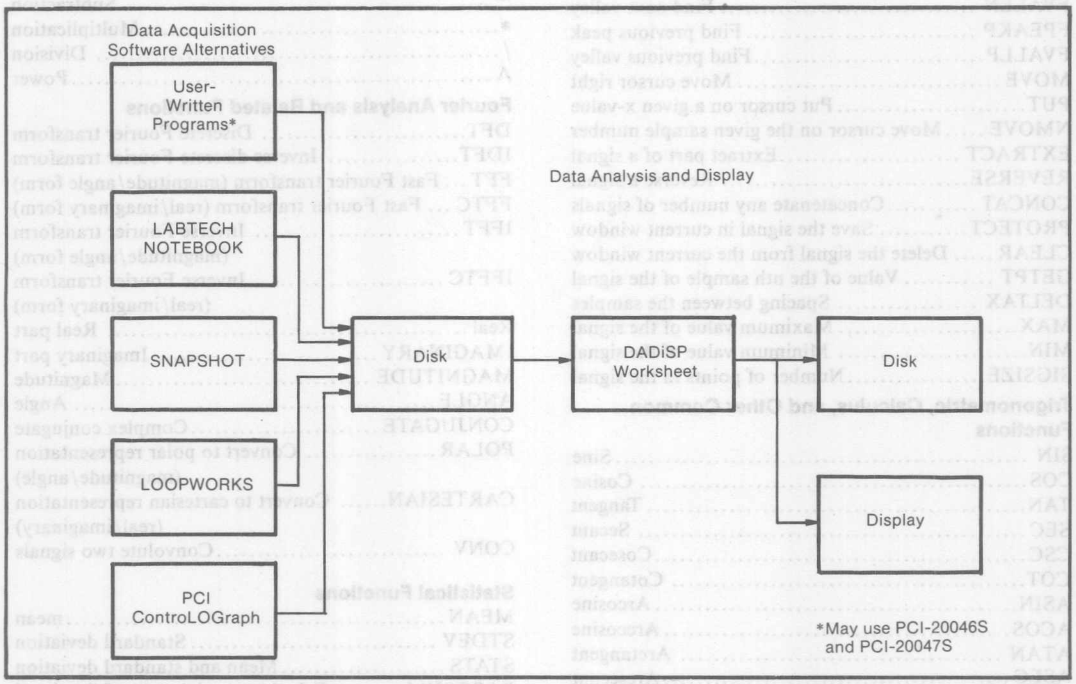


FIGURE 2. Block Diagram Illustrating use of DADiSP with other PCI Software Products.

DADiSP WORKSHEET FUNCTIONS

Waveform Generation

GSIN .... Sine wave
GCOS .... Cosine wave
GTAN .... Tangent curve
GSEC .... Secant curve
GCSC .... Cosecant curve
GCOT .... Cotangent
GASIN .... Arcsine
GACOS .... Arcosine
GATAN .... Arctangent
GASEC .... Arcsecent
GACSC .... Arccosecant
GACOT .... Arccotangent
GSINH .... Hyperbolic sine
GCOSH .... Hyperbolic cosine
GTANH .... Hyperbolic tangent
GSECH .... Hyperbolic secant
GCSCH .... Hyperbolic cosecant
GCOTH .... Hyperbolic cotangent
GASINH .... Hyperbolic arcsine
GACOSH .... Hyperbolic arccosine

SOFTWARE APPLICATIONS PACKAGES
PCI-20067S-1 DATA SHEET

11

GATANH .... Hyperbolic arctangent
GASECH .... Hyperbolic arcsecant
GACSCH .... Hyperbolic arccosecant
GACOTH .... Hyperbolic arccotangent
GSQRWAVE .... Square wave
GTRIWAVE .... Triangle wave
GEXP .... Exponentiation
GLOG .... Log
GIN .... Natural log
GLOGI0 .... Log base ten
SGQRT .... Square root
GLINE .... Any line

Peak Analysis, Signal Editing, and Related Operations

FMIN .... Find minimum value
FMAX .... Find maximum value
FPEAK .... Find first peak
FVALL .... Find first valley
FPEAKN .... Find next peak
FVALLN .... Find next valley
FPEAKP .... Find previous peak
FVALLP .... Find previous valley
MOVE .... Move cursor right
PUT .... Put cursor on a given x-value
NMOVE ..... Move cursor on the given sample number
EXTRACT .... Extract part of a signal
REVERSE .... Reverse a signal
CONCAT .... Concatenate any number of signals
PROTECT.......... Save the signal in current window
CLEAR ..... Delete the signal from the current window
GETPT .... Value of the nth sample of the signal
DELTAX .... Spacing between the samples
MAX .... Maximum value of the signal
MIN .... Minimum value of the signal
SIGSIZE .... Number of points in the signal

Trigonometric, Calculus, and Other Common Functions

SIN .... Sine
COS .... Cosine
TAN .... Tangent
SEC .... Secant
CSC .... Cosecant
COT .... Cotangent
ASIN .... Arcosine
ACOS .... Arccosine
ATAN .... Arctangent
ASEC .... Arcsecant
ACSC .... Arccosecant
ACOT .... Arccotangent
SINH .... Hyperbolic sine
COSH .... Hyperbolic cosine
TANH .... Hyperbolic tangent
SECH .... Hyperbolic secant
CSCH .... Hyperbolic cosecant
COTH .... Hyperbolic cotangent
ASINH .... Hypberolic arcsine
ACOSH .... Hyperbolic arccosine
ATANH .... Hyperbolic arctangent
ASECH .... Hyperbolic arcsecant
ACSCH .... Hyperbolic arccosecant
ACOTH .... Hyperbolic arccotangent
EXP ... Exponentiation (to the power of e = 2.7182)
LOG .... Log
LN .... Natural log
LOG 10 .... Log base 10
SQRT .... Square root
ABS .... Absolute value
SINC .... Sinc function (sin[x]/x)
FLOOR .... Round down (to next lower integer)
CEILING .... Round up (to next higher integer)
DERIV .... Derivative
INTEG .... Integral
LDERIV .... Left derivative
RDERIV .... Right derivative
+ .... Addition
− .... Subtraction
* .... Multiplication
/ .... Division
∧ .... Power

Fourier Analysis and Related Functions

DFT .... Discrete Fourier transform
IDFT .... Inverse discrete Fourier transform
FFT ... Fast Fourier transform (magnitude/angle form)
FFTC ... Fast Fourier transform (real/imaginary form)
IFFT .... Inverse Fourier transform (magnitude/angle form)
IFFTC .... Inverse Fourier transform (real/imaginary form)
Real .... Real part
IMAGINARY .... Imaginary part
MAGNITUDE .... Magnitude
ANGLE .... Angle
CONJUGATE .... Complex conjugate
POLAR .... Convert to polar representation (magnitude/angle)
CARTESIAN...... Convert to cartesian representation (real/imaginary)
CONV .... Convolute two signals

Statistical Functions

MEAN .... mean
STDEV .... Standard deviation
STATS .... Mean and standard deviation
PARTSUM .... Calculate partial sums of the signal
AMPDIST .... Calculate amplitude distribution
MOVAVG .... Calculate the moving average
LINREG .... Linear regression of one signal
LINREG2............ Linear correlation of two signals
SUMS .... Sum any number of signals
AVG .... Average any number of signals

BURR-BROWN®

PCI-20068S-1

SNAPSHOT STORAGE SCOPE
Software for Data Acquisition and Display

FEATURES

- Completely menu driven, no programming required
- Works with IBM PC, PC/XT, PC/AT and other compatibles
- Emulates a digital oscilloscope
- Supports the PCI-20000 Hardware System
- Records up to 16 channels, single ended or differential
- Displays up to eight channels concurrently
- Replays recorded data from disk for visual analysis
- Can magnify display in both vertical and horizontal directions
- Dual cursor displays both time and voltage values (in engineering units)
- Internal or external pre and post trigger capabilities
- Provides both analog and digital excitation signals
- Interfaces to SNAP-CALC, DADiSP and Lotus 1-2-3 for added analysis functions
- Provisions for anti-alias filter on termination panels

DESCRIPTION

SNAPSHOT STORAGE SCOPE is an integrated, menu driven software package specifically designed for data acquisition, graphic display and data storage. The package is suitable for use with all IBM PC, PC/XT, PC/AT and other compatible computers. SNAPSHOT STORAGE SCOPE automates the operation of the PCI-20000 analog input system, recording up to sixteen single ended or differential channels. Start of the acquisition process can be internally or externally triggered on an analog or digital signal with user defined slope (+ or −) and amplitude (−10V to +10V). In addition, external triggering is available from a user-defined eight-bit digital pattern. From one to eight channels can be displayed at the same time. Recorded data can be recalled from disk for further visual anlaysis aided by convenient cursor, offset and zoom functions.

The display screen is divided into four sections: operator information, labels, waveform display area and user options (see Figure 2). The operator information provides current setup values including channel sensitivity (volts/division), channel gain, sweep time, time per X-division, number of data points collected for each channel and the sampling time between points.

Each channel displayed has ten grid marks in each direction providing the type of graticule background typically found on an oscilloscope. Channel default labels can be modified along the left hand margin so that the waveforms can be easily identified.

User options are listed along the bottom line of the display. Each option has a short label describing the command, which can be executed by pressing the specific function key assigned to it. Minimal keystrokes are required to issue commands, specify parameters and select files.

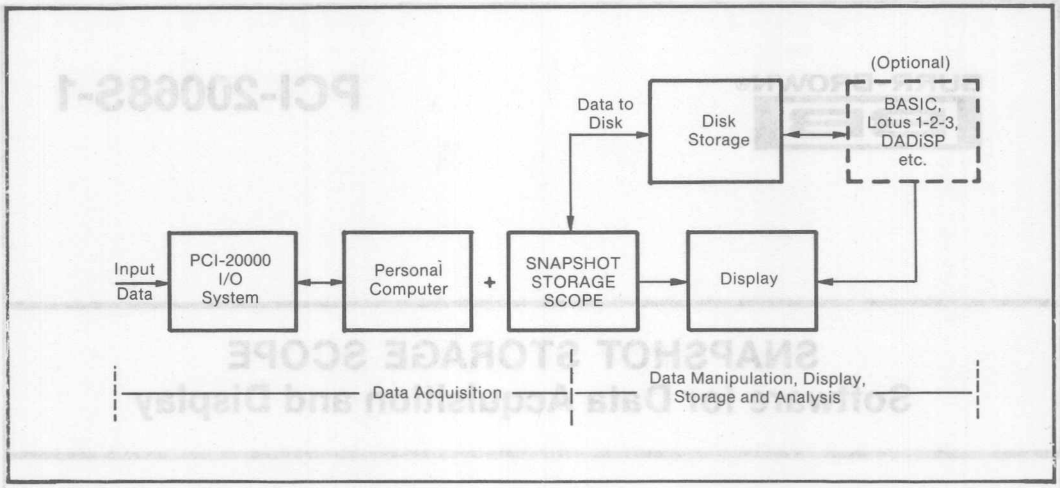


FIGURE 1. Illustration of the Use of SNAPSHOT STORAGE SCOPE in a PC-Based Data Acquisition and Analysis System.

SNAPSHOT STORAGE SCOPE is completely menu driven, requiring no programming skills or the memorization of commands. There are two types of menus: Command Menus and Setup Menus. Command menus are displayed on the bottom line of the display screen and are selected by pressing the function key that corresponds to the listed command. Commands can be used to either perform an action, such as data collection, or to select a new menu. The Setup Menus provide a list of the parameters that may be changed by the operator. There are two types of Set Up Menus: Configuration and Parameter menus. The Configuration Menus provide entries related to the setup of the hardware and software environments. The Parameter Menus are used to review and change the values that control data collection and graphic display of data on the screen. Invalid entries are detected immediately after entry by SNAPSHOT STORAGE SCOPE and a request to re-enter the parameter value is displayed. Examples of Parameter Menus for setting up the data acquisition and viewing parameters are shown in Figures 3 and 4.

Additional functionality can be gained by combining SNAPSHOT STORAGE SCOPE with other products such as SNAP-CALC, SNAP-FFT, DADiSP (PCI-20067S-1). Data can be ported to DADiSP for reduction, signal analysis and manipulation.

Please refer to the PCI-20000 hardware Data Sheet section for a complete description of the many available system components.

MENU OPTIONS

MAIN—The Main menu selects the major mode of operation. It may also set up the configuration and data collection/display parameters.

TRACE—The Trace menu controls data collection, viewing and recording.

REPLAY—The Replay menu controls the reading and display of previously recorded data files.

LOAD/SAVE—The Save parameter menu stores the current values being used for data collection and display. These parameters are assigned a file name and stored on disk so that they can be recalled at a later time. Multiple parameter files can be saved on disk. The Load Parameter menu retrieves previously recorded set up values and uses them to control the data collection and display.

CONFIGURATION—The Configuration menu is used to review and change the parameters that control the hardware and software setup.

PARAMETER—The Parameter menu is used to review and change the parameters that control data collection and display (see Figures 3 and 4).

LIST—The List menu displays a numeric listing of the sampled data points on the display screen (see Figure 5).

CURSOR—The Cursor menu allows the operators to select a cursor for any displayed trace. The cursor may be moved forward and backward along the trace while the value at the cursor position is displayed on the screen. The cursor function has the capability to display both relative and absolute values for both voltage and time (see Figure 6).

XZOOM—The zoom option may be selected from the Trace and Replay menus to provide a magnified display of a portion of the collected data.

SPECIFICATIONS—SNAPSHOT STORAGE SCOPE—PCI-20068S-1

PARAMETER	CONDITIONS		SPECIFICATION
Compatibility	Computers		IBM PC, XT, AT
	Data Acquisition Hardware		PCI-20000 System
Analog Input			16 channels max
Channels	PCI-20019M-1	(SE)	8
	with PCI-20031M-1	(SE)	16
Signal Gain	PCI-20019M-1		1
Resolution			12 Bits
Full Scale Range	PCI-20019M-1 Module		±2.5V to ±10V
Accuracy			.01% reading, ±1 bit
Output Signals	Control and Excitation		Analog and digital
Speed	PC/XT		800/second
	PC/AT		3400/second
Analog			
Channels			1
Output Range	Voltage (12-bit resolution)		±10V
	Load Current		±5mA
Digital			
Channels	TTL Levels		8 bits
Acquisition			
Sample Rate	Minimum		7/hour
	PCI-20019M-1, PC/XT		89K/second
	PC/AT		46K/second
Data Points	Maximum, Total for All Channels		32,000
Channel Display			1 to 8
Triggering	Free Run or Triggered		Analog or digital
	Analog, Internal/External		
	Trigger Level		±70mV to ±10V
	Digital, External		Up to 8 bits
Sweep Time	Maximum		200 days
	Minimum		32K/sample rate
Display			
Interface	IBM or Compatible		CGA, EGA, Hercules
Format			Cursor Readout, Plot, Tabular
Cursor	User defined units		Absolute, Relative
Zoom	X direction		±100 times
	Y direction		±500 times
Offset	Y direction only		±10V
Data Storage	Data formats		ASCII, Exponential Hex, Compacted Hex

REQUIRED PC HARDWARE

IBM PC, PC/XT, PC/AT or true compatible
256K RAM, minimum
IBM compatible graphics board
Graphics monitor
One disk drive
Coprocessor is recommended

PCI-20000 SUPPORTED HARDWARE

PCI-20041C-3	Carrier Board with Digital I/O and DMA
PCI-20019M-1	Fast A/D Module (12 bits)
PCI-20031M-1	Expander Module
PCI-20020M-1	Trigger Module
PCI-20003M-2	Analog Output Module

Termination Panels, Cables and Accessories

PCI-20057T-1	Analog Termination Panel
PCI-20011T-1	Digital Termination Panel
PCI-20042T-1	Isolated Analog Signal Conditioner (four channel)
PCI-20043T-1	Isolated Expansion Signal Conditioner (four channel)
PCI-20044T-1	Active Analog Signal Conditioner (four channel)
PCI-20045T-1	Active Expansion Signal Conditioner (four channel)
PCI-20048T-1	Isolated Digital Termination Panel
PCI-20012A-1	Analog Cable
PCI-20013A-1	Digital Cable
PCI-20038A-1/2	±15V Power Supply (for PCI-20042T-45T)
PCI-20028A-2	Strain Relief Bracket
PCI-20029A-1	Rack Mount Enclosure (room for four panels)
PCI-20051A-1	Rack Mount Enclosure (for PCI-20048T-1)
PCI-20052A-1	Cover (for PCI-20051A-1)
PCI-1100 Series	Series Opto Modules (for PCI-20048T-1)

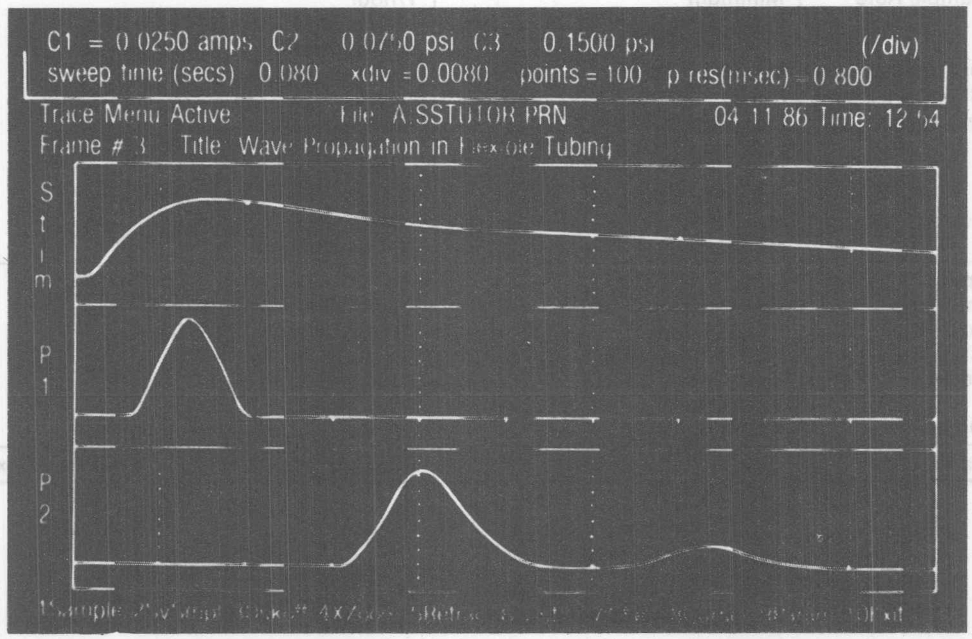


FIGURE 2. Screen Display—Shows sensitivities, number of points, title, date, time, labels and waveforms for one to four channels, and menu for user options. See data as it is acquired.

HARDWARE SYSTEM #1

High speed, single ended (with or without digital isolation):

PCI-20041C-3	Carrier Board with Digital I/O and DMA
PCI-20019M-1	Fast A/D Module (12 bits)
PCI-20020M-1	Trigger Module

Options

PCI-20031M-1	Expander Module
—or—	
PCI-20003M-2	Analog Output Module
PCI-20057T-1	Analog Termination Panel
PCI-20011T-1	Digital Termination Panel
—or—	
PCI-20048T-1	Isolated Digital Termination Panel
PCI-1100 Series	Opto Modules, for PCI-20048T-1 (up to 16)
PCI-20012A-1	Analog Cable (three required)
PCI-20013A-1	Digital Cable
PCI-20028A-2	Strain Relief Bracket
PCI-20029A-1	Rack Mount Enclosure (room for four Panels)
PCI-20051A-1	Rack Mount Enclosure (for PCI-20048T-1)
PCI-20052A-1	Cover (for PCI-20051A-1)

HARDWARE SYSTEM #2

High Speed, Differential Input (with or without Analog and Digital Isolation):

PCI-20041C-3	Carrier Board with Digital I/O and DMA
PCI-20019M-1	Fast A/D Module (12 bits)
PCI-20020M-1	Trigger Module

Options

PCI-20031M-1	Expander Module
—or—	
PCI-20003M-2	Analog Output Module
PCI-20057T-1	Analog Termination Panel
PCI-20042T-1	Isolated Analog Signal Conditioner (four channel)
PCI-20043T-1	Isolated Expansion Signal Conditioner (four channel)
PCI-20044T-1	Active Analog Signal Conditioner (four channel)
PCI-20045T-1	Active Expansion Signal Conditioner (four channel)
PCI-20038A-1/-3	±15V Power Supply (for PCI-20042T-45T)
PCI-20011T-1	Digital Termination Panel
—or—	
PCI-20048T-1	Isolated Digital Termination Panel
PCI-1100 Series	Opto Modules, for PCI-20048T-1 (up to 16)
PCI-20012A-1	Analog Cable (three required)
PCI-20013A-1	Digital Cable
PCI-20028A-1	Strain Relief Bracket
PCI-20029A-1	Rack Mount Enclosure (room for four Panels)
PCI-20051A-1	Rack Mount Enclosure (for PCI-20048T-1)
PCI-20052A-1	Cover (for PCI-20051A-1)

```
                    PARAMETER MENU

 *******  Data Acquisition Parameters  *******

  ( 1 )  Sample Rate:      1000.00 Hz;   Points/Channel: 250
  ( 2 )  Gain :     1.0
  ( 3 )  Sample Channels:   1 -   4
  ( 4 )  Sweep Time:      0.250  Seconds
  ( 5 )  Trigger:  Digital   Bit 1
  ( 6 )  Outputs:  Non-sampling:        0000 (D)       0.00(A)
                       Sampling:        0000 (D)       0.00(A)

 *******  Viewing Parameters  *******

  ( 7 )  Unit Conversion Parameters and Labels
  ( 8 )  Y Display Parameters
  ( 9 )  Channels Displayed:   1[P1]  2[P2]  3[Q1]  4[Q2]
 ( 10 )  Number of Windows:  1
 ( 11 )  Step Rate for Plotting:    1
 ( 12 )  X Display Range:   0.00  to 1.00
 ( 13 )  Data Stored:  Averaged

 Enter selection:
```

FIGURE 3. Parameter Menu—Each command is only one or two keystrokes. No programming experience required.

```
Enter Sample Rate
     Max Range [0.000470 to 50000]    Dependent Range [39.94 to 12500.00]
     Current 1000.00 Hertz




     NOTE:   Sample Rates outside of the Dependent Range require a different
             Sweep Time or Number of Channels.

     NOTE: Actual Sample Rate may differ from value entered
           because of A/D board timer limitations.
```

FIGURE 4. Sub-Menu Displays range of values allowable and value presently selected.

Hit esc key to stop listing

FRAME # 1 CHANNEL # 2 UNITS: amps

1=	-0.000610	2=	0.000000	3=	-0.000610	4=	0.000000
5=	-0.000610	6=	0.000000	7=	-0.000610	8=	0.000000
9=	0.023193	10=	0.125732	11=	0.270996	12=	0.405884
13=	0.524292	14=	0.595703	15=	0.607300	16=	0.562744
17=	0.476074	18=	0.358887	19=	0.223999	20=	0.112915
21=	0.011597	22=	0.001221	23=	-0.002441	24=	0.000000
25=	-0.002441	26=	0.002441	27=	-0.003052	28=	0.001221
29=	-0.005493	30=	0.001221	31=	0.000000	32=	0.002441
33=	0.000000	34=	-0.000610	35=	-0.003052	36=	-0.000610
37=	0.000000	38=	-0.000610	39=	0.000000	40=	-0.000610
41=	0.000000	42=	-0.000610	43=	0.000000	44=	-0.000610
45=	0.000000	46=	-0.000610	47=	0.000000	48=	-0.000610
49=	0.000000	50=	-0.000610	51=	0.000000	52=	-0.000610
53=	0.000610	54=	-0.001831	55=	0.001831	56=	-0.000610

Average = 0.0355 Minimum = -0.0482 Maximum = 0.6073

Select function key

1:Ch-1 2:Ch-2 3:Ch-3 4 5 6 7 8 9 10Exit

FIGURE 5. Data Table—Lists data point numbers and values.

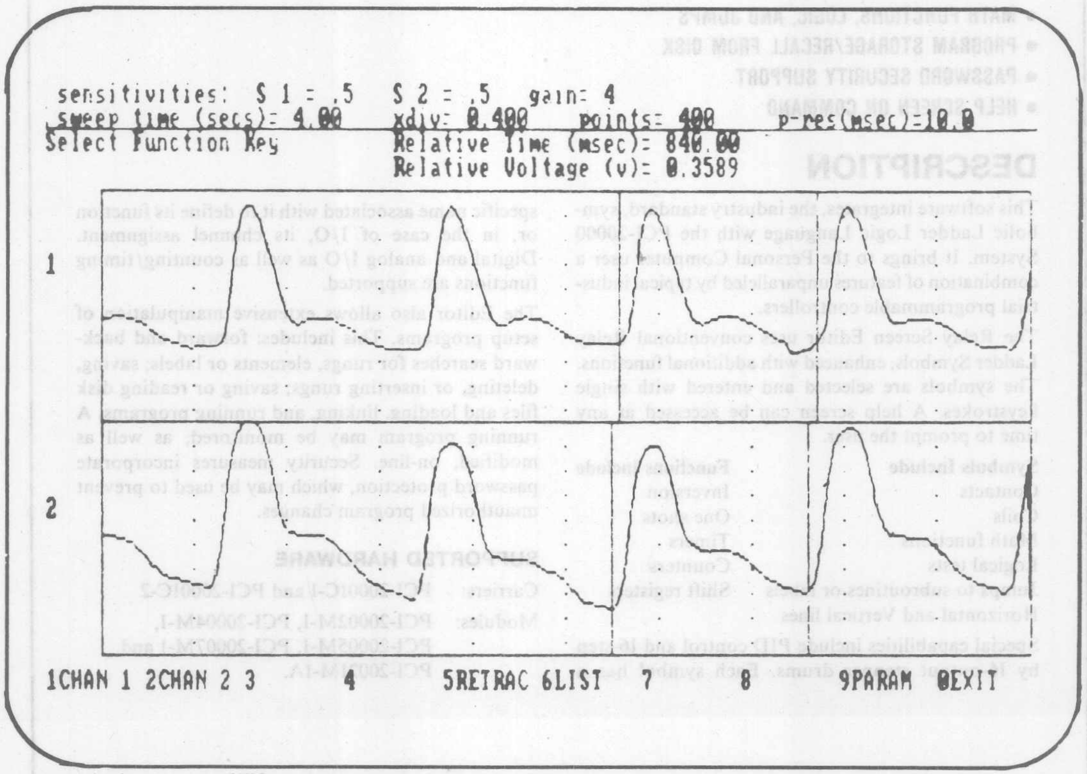


FIGURE 6. Cursor Control—Displays either relative or absolute time and voltage values.

BURR-BROWN®

PCI-20073S-1

ADVANCE INFORMATION
Subject to Change

Relay Ladder Logic RD1000/PC Software For PROCESS MONITORING AND CONTROL

FEATURES

- RELAY SYMBOL/SCREEN EDITOR
- ON-LINE RELAY/COIL MONITORING
- LOAD AND TEST RUNG BY RUNG
- ANALOG I/O, DIGITAL I/O AND COUNTING
- PID AND DIGITAL (ON/OFF) CONTROL
- MATH FUNCTIONS, LOGIC, AND JUMPS
- PROGRAM STORAGE/RECALL FROM DISK
- PASSWORD SECURITY SUPPORT
- HELP SCREEN ON COMMAND

APPLICATIONS

- PROCESS MONITORING
- BATCH PROCESS CONTROL
- CONTINUOUS PROCESSING
- ENERGY MANAGEMENT

DESCRIPTION

This software integrates, the industry standard, symbolic Ladder Logic Language with the PCI-20000 System. It brings to the Personal Computer user a combination of features unparalleled by typical industrial programmable controllers.

The Relay Screen Editor uses conventional Relay Ladder Symbols, enhanced with additional functions. The symbols are selected and entered with single keystrokes. A help screen can be accessed at any time to prompt the user.

Symbols Include	Functions Include
Contacts	Inversion
Coils	One shots
Math functions	Timers
Logical tests	Counters
Jumps to subroutines or labels	Shift registers
Horizontal and Vertical lines	

Special capabilities include PID control and 16-step by 16-output stepper drums. Each symbol has a specific name associated with it to define its function or, in the case of I/O, its channel assignment. Digital and analog I/O as well as counting/timing functions are supported.

The Editor also allows extensive manipulation of setup programs. This includes: forward and backward searches for rungs, elements or labels; saving, deleting, or inserting rungs; saving or reading disk files and loading, linking, and running programs. A running program may be monitored, as well as modified, on-line. Security measures incorporate password protection, which may be used to prevent unauthorized program changes.

SUPPORTED HARDWARE

Carriers: PCI-20001C-1 and PCI-20001C-2

Modules: PCI-20002M-1, PCI-20004M-1, PCI-20005M-1, PCI-20007M-1 and PCI-20021M-1A.

Additional Software
Available from Others

There is a large and growing body of high-performance software available for use with the PCI-20000 System. Much of this software has been developed by organizations who specialize in a particular type of software. On the following pages, we feature vendor-supplied information regarding selected software packages for your consideration. Each of these software packages interfaces to products within the Burr-Brown PCI-20000 product line, and each is optimized for a particular application area. We invite you to consider them for your application.

For further information or for purchase of any of these software products, you should contact the vendor directly. Any product warranties will be provided by the vendor, not by Burr-Brown. The product information in this section is based on material supplied by the vendors. Burr-Brown assumes no responsibility for omissions or inaccuracies.

In addition to the products that follow in this section, products manufactured by Macmillan and Lotus are often of interest to software users:

1. **ASYST** (Data Acquisition, Analysis, and Graphics)

   Description: ASYST is a scientific software package which allows the acquisition, analysis and graphic display of real-world data. Module 1 provides graphics and statistics functions. Module 2 provides various mathematical and curve fitting functions. The Burr-Brown PCI-20046S-4 software package interfaces the PCI-20000 System to ASYST.

   ASYST Software Technologies
   100 Corporate Woods
   Rochester, NY 14623
   (716) 272-0070

2. **Lotus 1-2-3** (Spreadsheet with Graphics, see Section 9 for example of use with PCI-20000 System)

   Lotus Development Corporation
   161 First Street
   Cambridge, MA 02142

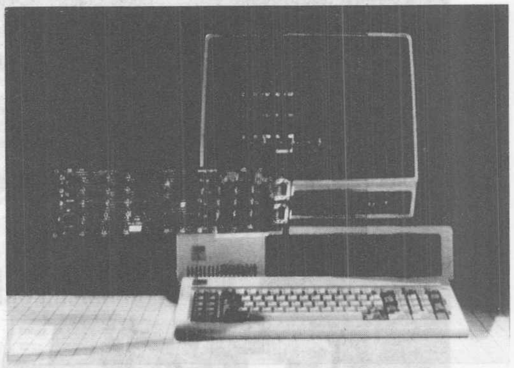
CODAS

Hardware and Software for PC-Based Oscillograph

Available from: **Dataq Instruments, Inc.** • 825 Sweitzer Ave. • Akron OH 44311, USA
Telephone: 216-434-4284 • Telex: 650-281-7028

FEATURES

- **Continuous data streaming to floppy or hard disk at rates up to 2000 (PC, PC/XT) and 5000 samples per second (PC/AT).**
- **Simultaneous real-time data plotting directly on the computer's monitor through the unique Graphics Accelerator Card.**
- **Two selectable real time data display modes: continuous smooth scroll plotting (like chart recorder), and triggered sweep (like oscilloscope).**
- **Supports 16 channels of data acquisition.**
- **Supports eight channels of display.**
- **Data files compatible with the most popular analysis packages such as RS/1, Lotus 1-2-3, and DADiSP.**
- **Works with PCI-20000 Data Acquisition System.**
- **No programming ever required.**



DESCRIPTION

The Computer-based Oscillograph and Data Acquisition System (CODAS) is a combination hardware/software package allowing continuous data throughput to floppy or hard disk while maintaining a real time display directly on the host computer's monitor.

REAL TIME DATA ACQUISITION TO DISK

Continuous streaming of acquired data to the selected disk may be enabled or disabled by a simple keystroke. The quantity of data to be acquired is not limited by the size of reserved semiconductor memory, but rather by the size of the target disk. Such an approach allows real-time data acquisition capacities in the tens of megabytes if necessary, while still maintaining the ability to acquire small or moderate sized data sets.

REAL-TIME DISPLAY

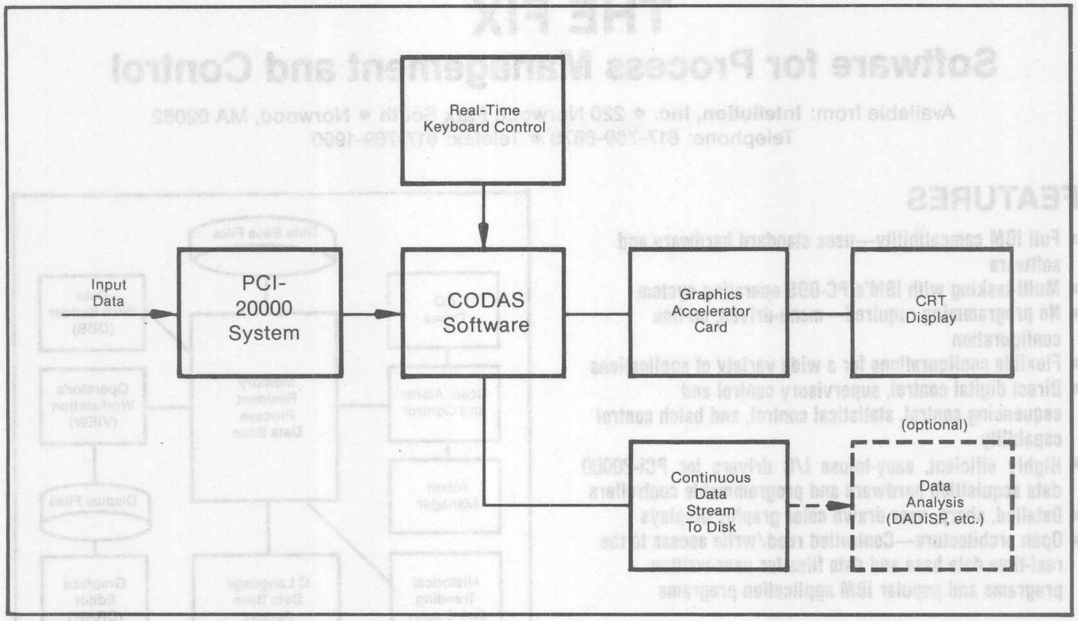
While data is being acquired to disk, a real-time plot precisely describing the acquired data is enabled for display directly on the computer's monitor through the provided Graphics Accelerator Card. Waveforms generated by each enabled A-D channel are displayed in a unique continuous smooth scroll format. This plotting technique provides the ultimate in waveform display continuity and exactly mimics the continuous plot nature of a chart recorder. Alternatively, a triggered sweep (oscilloscope-like) display appearance may be enabled. Such a display mode readily adapts to higher frequency input signals by overcoming the natural limitation of the eye attempting to follow a fast-moving, scrolling graph. Up to eight channels may be simultaneously plotted on the computer's monitor, and the real-time display is always active and independent of disk storage activity.

POST-ACQUISITION SUPPORT

Included with the CODAS package is waveform analysis software allowing the review and qualification of data previously acquired to disk. Specific waveform voltages may be measured as well as timing intervals on the same or different channels. Data sets of any size may be reviewed on the computer's monitor using the smooth scroll waveform presentation. Scrolling through the data set may be done in either a positive or negative time direction in a manner similar to reviewing the paper record of a chart recorder. A high waveform scroll speed of over 4000 points per second (using an IBM PC/AT) allows even the largest data file to be reviewed in seconds—from beginning to end. A copy and paste utility is included that allows any waveform group to be copied and pasted to one or more separate data files on floppy or hard disk for further analysis and hard copy using such popular analysis packages as Lotus 1-2-3, RS/1, and DADiSP.

CODAS's logical keyboard command structure makes data acquisition an intuitive process. All interaction with the real-time data acquisition process is accomplished through easy-to-remember keystrokes.

Since the real time display is always active (independent of disk storage operations), any adjustments you make are instantly reflected on the real time screen. For example, increasing the sample rate will expand the real time plot of displayed waveforms as the number of samples within a given unit of time increases. Likewise, up- or down-scaling a selected channel will instantly expand or contract that channel's waveform in the amplitude direction. And, all keyboard functions are active during disk acquisition, allowing you to make real-time display adjustments on the fly.



CODAS Functional Block Diagram.

SPECIFICATIONS

CODAS

Supported monitors	IBM monochrome, IBM RGB color, Enhanced Graphics Display, and compatibles.
Supported video communications	Separate sync and video (noncomposite).
Supported host video cards	Any compatible with the above monitors, including the IBM Enhanced Graphics Adaptor card.
Waveform pixel resolution	Selectable 640 × 350, 640 × 200, 400 × 350, 400 × 200 pixels.
Waveform display modes	Selectable continuous smooth scroll, or triggered sweep.
Trigger conditions for triggered sweep waveform display mode	± slope selectable, selectable Trigger level and source.
Number of waveform display formats	Ten.
Number of acquired channels	16
Number of waveform display channels	Eight.
Maximum real time display throughput rate	2kHz (PC/XT), 5kHz (PC/AT).
Maximum continuous throughput to disk acquisition rate	2kHz (PC/XT), 5kHz (PC/AT).
Sample rate ranges (step)	1–10 (1), 10–100 (10), 100–1000 (100), 1000–5000 (1000) samples per second.

GRAPHICS ACCELERATOR CARD

See Dataq Instruments' WFS-200PC description in this section for further information.

Vector draw rate	15,000 vectors per second.
Pixel draw rate	5.25 x 10^6 pixels per second.
Number of waveform channels	Eight.
Waveform display modes	Continuous smooth scroll (horizontal pan), selectable left-to-right or right-to-left. Triggered sweep (retrace plotting upon valid trigger conditions).

CODAS MINIMUM COMPUTER REQUIREMENTS

- IBM PC/XT/AT or compatible.
- 128 Kbytes of memory.
- One DSDD or SSDD floppy disk drive—hard disk optional.
- Any IBM-compatible graphics or monochrome display and printer adaptor card (graphics capability not required).
- IBM-compatible monochrome or color monitor (noncomposite).
- PC-DOS 2.0 or greater.

BURR-BROWN PCI-20000 HARDWARE SUPPORTED

PCI-20001C-1.......................... Carrier Board
PCI-20001C-2.......................... Carrier Board
PCI-20041C-2.......................... Carrier Board
PCI-20002M-1........ 12-Bit Data Acquisition Module

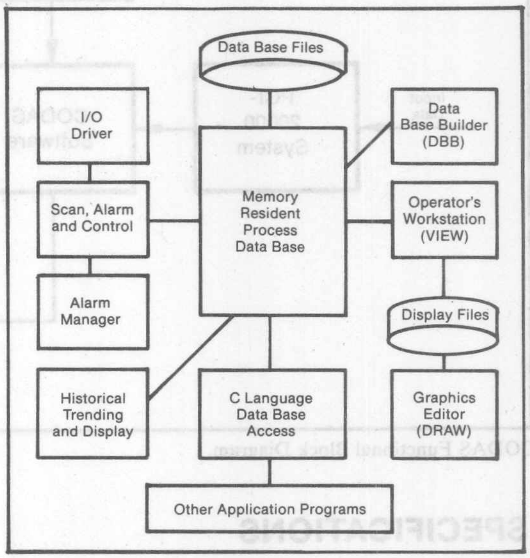
THE FIX

Software for Process Management and Control

Available from: **Intellution, Inc.** • 220 Norwood Park South • Norwood, MA 02062
Telephone: 617-769-8878 • Telefax: 617-769-1990

FEATURES

- **Full IBM compatibility—uses standard hardware and software**
- **Multi-tasking with IBM's PC-DOS operating system**
- **No programming required—menu-driven, on-line configuration**
- **Flexible configurations for a wide variety of applications**
- **Direct digital control, supervisory control and sequencing control, statistical control, and batch control capability**
- **Highly efficient, easy-to-use I/O drivers for PCI-20000 data acquisition hardware and programmable controllers**
- **Detailed, sharp, user-drawn color graphic displays**
- **Open architecture—Controlled read/write access to the real-time data base and data files for user-written programs and popular IBM application programs**



DESCRIPTION

THE FIX provides comprehensive data acquisition, process management and control functions for the IBM Industrial and Personal Computers. Familiar instrumentation blocks can be connected to PC-based I/O hardware, such as the Burr-Brown PCI-20000 series, or to external hardware, such as programmable controllers.

THE FIX also gives you interactive menus for configuring your system. With these blocks, you can construct complex control networks, *with no programming necessary.* In addition, you can build, modify, and tune your process control strategy *on-line* without disturbing the system.

COLOR GRAPHICS

THE FIX features two great ways for you to create your own interactive process graphic displays in full color: character graphics and pixel graphics. These choices give you all the flexibility you need to give your system the highest performance and best-looking operations and management displays available.

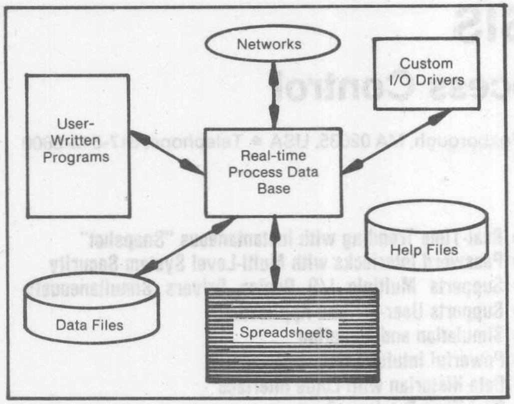
The standard method for building displays with THE FIX is with character graphics. Character graphic displays draw quickly because the symbols are pre-configured to go together well. An EPROM chip supplied with the system gives you over 128 different instrumentation symbols. These include symbols for horizontal and vertical valves, solenoid and modulated valves, pumps, pipe, ladder logic symbols, tank sides and bar graph segments.

The Pixel Graphics option provides the ultimate in flexibility and high resolution. Pixel displays are completely free format—you can create your own symbol set, too. A high resolution monitor and IBM's Enhanced Graphics Adapter with 256K of add-on memory (or equivalent) will give you full color displays with 640 by 350 pixel resolution.

OPEN ARCHITECTURE

Since THE FIX uses PC-DOS, its open architecture allows you to share real-time process data with other user-written programs and popular spreadsheets. There may also be times when you need to interface to special I/O devices. THE FIX provides a firm and secure foundation that makes customizing and adding functions easy.

The On-Line Spreadsheet Interface provides a mechanism for transferring data from THE FIX to other IBM application programs. With the OSI, you can develop a display that becomes your data file format. Then, on command, the data is written to a disk file that can be imported by a spreadsheet program or even transmitted to a host computer. The OSI also contains an "add-in" for Lotus Symphony™. With the add-in, you can read and write process data *in real time* between the Symphony spreadsheet and FIX Process Data Base.



HISTORICAL TRENDING

The Historical Trending Package provides an automatic and comprehensive means of sampling, storing and displaying process data over long periods of time. The Trend package provides data handling, display and exporting features. Data collection and display operations are performed *on-line* giving you maximum flexibility for process analysis.

With the Trend Display program, the operator can view up to eight points on a single time-value graph in full color. The Display program has a number of powerful features, including chart shift, engineering units scaling, pan (screen compression), and zoom (screen expansion). A movable time cursor provides an exact value/status readout of selected points.

CONFIGURATIONS

THE FIX is available in five different configurations that make it very easy to configure to fit your exact system needs.

All FIX system configurations feature the same easy-to-use menus, on-line PC-DOS functions and comprehensive operations that have made THE FIX an unbeatable system. Standard Features include:

- Familiar instrumentation blocks for data acquisition and control functions
- Nine-character tag names and 40-character descriptors
- Two-level on-line help screen available on a single keystroke
- Logging of all operator actions
- User-specified passwords and display security

1. SCADA Base-Line

The SCADA Base-line package provides a very economical way to get started using THE FIX. It is a full-featured package that performs:

- Data acquisition
- Analog and digital alarm detection and message generation
- Real-time trending
- On-line calculations
- User-drawn color graphics

In addition, the Base-line SCADA package will support the following FIX options:

- Historical Trending (includes THE FIX Shell)
- C Data Base Access
- On-line Report Generator
- On-line Spreadsheet Interface
- Graphic Printing Program

2. SCADA Expanded

The SCADA Expanded package forms the foundation for a full-sized FIX system. It contains advanced data handling and I/O communications functions. It can also be *upgraded* with any of the powerful and popular FIX system add-on packages:

- Direct Digital Continuous Control
- Statistical Process Control
- Batch and Sequencing Control

3. Continuous Control

The Control package add-on turns the SCADA Expanded system into a full, direct-digital-control system. Traditional control blocks can be chained together to form complex control loops. The Control package features on-line loop tuning, cascade capability and loop phasing. In addition to ratio and PID control, the Control package contains an on-off controller for heating and cooling operations, as well as a Drum Sequencer for simple batch and sequential applications.

4. Statistical Process Control

The Statistical Process Control package provides automatic or manual sampling of the process, as well as six types of *on-line alarming* based on statistical analysis. THE FIX SPC package is unique in that it can perform supervisory control on closed-loop systems.

5. Batch Control

The Batch Control package, when added to the SCADA Expanded system, gives you full continuous and batch control capability. In addition to the Continuous control blocks, the Batch package provides Boolean, Device, and Program blocks that support recipes and perform complex device handling and sequencing.

I/O DEVICE DRIVERS

THE FIX device drivers support the following Burr-Brown PCI-20000 I/O hardware:

PCI-20001C-2	Carrier Board with Digital I/O
PCI-20002M-1	16-Channel, 12-Bit Data Acquisition Module
PCI-20004M-1	32-Bit Digital I/O Module
PCI-20005M-1	32-Channel Analog Multiplexer Module
PCI-20007M-1	Counter, Timer, Pulse Generator Module
PCI-20021M-1	8-Channel, 12-Bit Analog Output Module

GENESIS

Software for Process Control

Available from: **Iconics, Inc.** • 132 Central Street, Suite #110 • Foxborough, MA 02035, USA • Telephone: 617-543-8600

FEATURES

- Icon Driven Creation of Control Strategies
- Free Hand Drawing of Process Displays
- Fast Single Key Access to Process Graphics
- True Real-Time Multitasking Operating System
- Fully Illustrated Manual and Quick Reference Card
- Self Documenting Control Strategies, Process Database and Wiring lists
- Expert Checker
- Over 40 Industry-Standard Control Algorithms
- Standard and User-Defined Graphics Symbol Libraries
- 20 User-Defined Function Keys for Fast Operator Response
- Comprehensive Alarm Management
- Real-Time Trending with Instantaneous "Snapshot"
- Password Interlocks with Multi-Level System Security
- Supports Multiple I/O Device Drivers Simultaneously
- Supports User-Written Applications
- Simulation and Modeling
- Powerful Intuitive User Interface
- Data Historian with Lotus Interface
- Real-Time Database Sorting
- Online Help
- Variable Scan Rates for Individual Control Algorithms
- No Programming Required
- Works with Burr-Brown PCI-20000 System Hardware

DESCRIPTION

GENESIS transforms the IBM PC into a multifunction CAD workstation for creating, simulating and implementing real-time data acquisition and process control strategies. Absolutely no programming is required. Simply point the mouse and click, it's that easy.

The complete GENESIS system includes Control & Graphics Builders (GENESIS-CGB) and a powerful Run-Time System (GENESIS-RT).

GENESIS-CGB

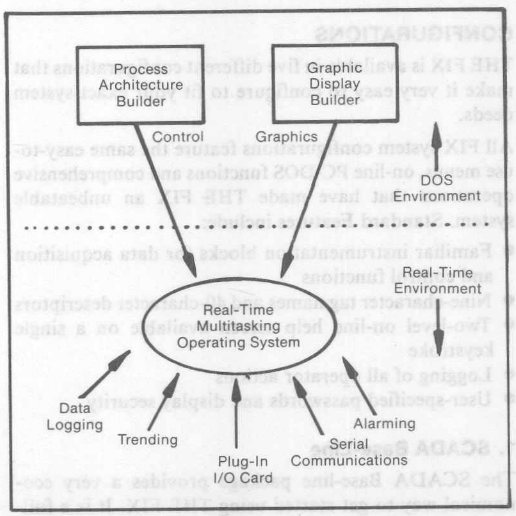
Process Architecture Builder

The Process Architecture Builder provides CAD tools to graphically create and edit data acquisition and control strategies. Using a mouse, the user selects, positions and connects icons representing control algorithms to create process control strategies. User-built process control strategies are self documenting and require no programming. Hard-copy reports of the process control database are available in graphical or tabular format. GENESIS resident Expert Checker detects and reports inconsistencies in the control strategy.

Graphics Display Builder

The Graphics Display Builder is an interactive graphics tool for creating dynamic process displays. Icon-based pick-and-place construction techniques are used to build and edit operator displays. Dynamic displays are created by graphically connecting display objects to the process database.

The GENESIS icon library contains over 60 standard symbols representing tanks, motors, pipes, valves, faceplates, indicators and instruments. New symbols and symbol libraries are easy to create and modify. Custom displays can be attached to individual function keys for fast and easy operator access during run-time. Displays can also be linked in a display hierarchy allowing the operator to quickly page through displays. A hard copy of any display can be generated with the print page icon.



GENESIS-RT

Run-Time System

The Run-Time System is a real-time process management and control function built around a powerful real-time multitasking operating system which is co-resident with DOS. The Operating System employs a prioritized preemptive scheduler with real-time interrupt support for communications and keyboard operations. The Run-Time System includes:

1. **Operator Interface**—An operator window provides the interface to all data acquisition and control functions. A split screen feature allows viewing displays in one window while simultaneously modifying parameters in another. Under password control, process parameters can be changed from the keyboard. Changes are immediately reflected on the operator window display.
2. **Comprehensive Alarm Management**—Alarms are prioritized and chronologically recorded. Alarms can be displayed via an Alarm Summary Display or printed. Alarm squelching quickly inhibits unimportant alarms. Unacknowledged alarms are displayed regardless of the current active display.
3. **Real-Time Trending**—Up to 20 variables may be trended simultaneously and any five can be displayed in one-, six- and 30-minute time intervals. A "snapshot" feature allows the operator to instantly capture any number of trend plots for later replay. A data zoom function allows the operator to magnify process curves. The Data Historian is available for longer term data storage and information management.
4. **System Security**—Selected parameters can be key-locked by one of three password security levels. The system may not be accidentally halted or re-booted from the keyboard.
5. **Real-Time Database Sorting**—The process database can be sorted in real-time. Up to five sort attributes are available. Sort attributes include: tag names, algorithm types and wild-card characters.
6. **Data Historian**—The Data Historian is an advanced long-term data collection and information management function. Up to 40 variables may be logged simultaneously on user selectable files of one-hour, eight-hour, 24-hour and one-week periods. Archived data is recorded to hard disk in a format compatible with Lotus or other popular spreadsheet programs. Each log entry is time- and date-stamped. Individual log titles identify each report.

I/O DEVICE DRIVERS

Genesis device drivers support the following Burr-Brown PCI-20000 I/O hardware:

PCI-20001C-2	Carrier board with digital I/O
PCI-20002M-1	16-channel, 12-bit data acquisition module
PCI-20005M-1	32-channel analog input expansion module
PCI-20007M-1	Counter/timer module
PCI-20021M-1	8-channel, 12-bit analog output module.

SPECIFICATIONS

PERFORMANCE
40 control loops per 1/4 second on 8MHz PC-AT.

DATABASE CAPACITY
400 points of a typical I/O mix.

CONTROL SCAN PERIOD
.25, .5, 1.0, 2.0, 6.0, 30.0 seconds.

SCREEN UPDATE RATE
1-second default, user configurable.

NUMBER OF DISPLAYS
Limited only by disk capacity.

CONTROL ALGORITHM LIBRARY
- I/O: AIN, AOUT, DIN, DOUT
- Control: PID, PD, INTEGRATOR, DGAP, AUTO/MANUAL/BIAS/RATIO STATION
- Logic: AND, OR, NAND, NOR, NOT, XOR, PULSE
- Calculation: LEAD-LAG, CHARACTERIZER, DEAD TIME, FILTER, SWITCH, LOG, e^x
- Selectors: HIGH, LOW, MEDIAN, AVERAGE, ALARM
- Math: ADD, MULT, DIV, SIN, COS, TAN, LOG, EXP
- Batch: RAMP, SEQUENCER, TOTALIZER, COUNTER, ONE-SHOT, TIME-ON, TIME-OFF
- Miscellaneous: TRIGGER, SIMULATION, UP TO FOUR USER WRITTEN ALGORITHMS

ALARM MANAGEMENT
- Alarm summary page
- Ten alarm priorities
- Selectable alarm priority
- Logic transition alarming
- High low limit alarms
- Rate of change
- Deviation alarming
- Report to printer
- Descriptive alarm messages
- Sequence of events alarm queuing
- Alarm squelch
- Predictive alarms
- Alarm acknowledgement

REAL-TIME TRENDING
- Trend up to 20 variables
- Display any five simultaneously
- Individual selection of zoom per "pen"
- Continuous sample averaging
- Selectable time base of one, six, or 30 minutes
- "Snapshot" feature captures data instantly for easy replay.

SYSTEM SECURITY
- Password protection on controller tuning parameters
- System may not be accidentally halted or rebooted from the keyboard.

DATA HISTORIAN
- Up to 40 variables simultaneously
- Data sampled every ten seconds
- averaged and stored every 30 seconds with time data stamp
- Individual report titles are automatically logged
- Data file compatibility with popular spreadsheet packages.

LABTECH CHROM

Software for Chromatography Analysis

Available from: **Laboratory Technologies Corp.** • 255 Ballardvale St. • Wilmington, MA 01887, USA
Telephone: 617-657-5400 • Telex: 989695

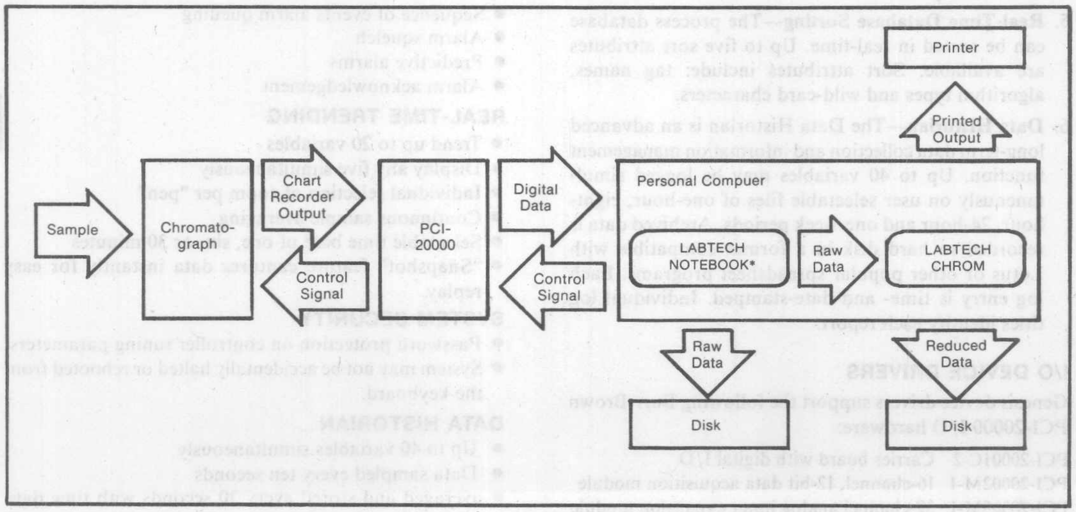
FEATURES

- **CHROM works as a unit with LABTECH NOTEBOOK* and the Burr-Brown PCI-20000 system for data acquisition.**
- **Accepts raw data directly from chromatograph data workstation.****
- **Multichannel operation—one PC can handle several chromatographs or channels.**
- **NOTEBOOK's foreground/background capability allows data collection to proceed simultaneously with analysis of data from a previous run.**
- **Raw data stored on permanent magnetic disk storage media for archiving—required by FDA regulations for good laboratory practices.****
- **User-friendly menu-driven system, requiring minimal training or operator expertise.**
- **Multisample/autosampler operation for unattended operation.****
- **Smoothes raw data by convolution and eliminates spikes in data.**
- **Streams data to disk continuously for multi-megabyte data storage capacity.****
- **Sophisticated baseline drawing algorithms, allowing accurate analysis of fused peaks, drifting baselines, etc.**
- **IBM PC compatible, can be used with the PCI-20000 hardware and LABTECH NOTEBOOK* software packages, or with software written in-house.**
- **Raw data reprocessing capability to fine tune analysis methods.**
- **Produces annotated graphical display on inexpensive dot matrix printers, eliminating the need for chart recorders.**
- **Automatically uses 8087/80287 floating point processor, if installed.**
- **Provides detailed results report, with peak area, peak start, peak maxima, peak end—provides for follow-on analysis, display or validation.**
- **Interfaces to popular systems like 1-2-3, Symphony, and RS/1 for further analysis.**

DESCRIPTION

LABTECH CHROM is a software program that turns an IBM personal computer into a chromatography integrator and data archive. It accepts raw chromatographic data and method parameters as its input, and it produces chromatographic peak areas and retention times as output. It works with LABTECH NOTEBOOK*, Laboratory Technologies' data acquition and control package.

LABTECH CHROM provides a basic integration capability for use as a system building block, or for standalone composition determination. It is oriented toward



*LABTECH NOTEBOOK is available as PCI-200040S-1.
** Features available when LABTECH CHROM is used with LABTECH NOTEBOOK.

production capillary gas chromatography or liquid chromatography, in which the chromatographic runs may contain many thousands of data points.

With the LABTECH CHROM approach, raw data is archived on permanent storage. This data is always available for reprocessing with different analysis methods. A complete trail of the analyses is maintained on the permanent disk storage media. This trail conforms to good laboratory practices, and is suitable for scrutiny by federal and state agencies such as the Food and Drug Administration and the Environmental Protection Agency.

CHROM is extremely well suited for unattended operation with autosampling devices that perform continuous or semi-continuous analyses.

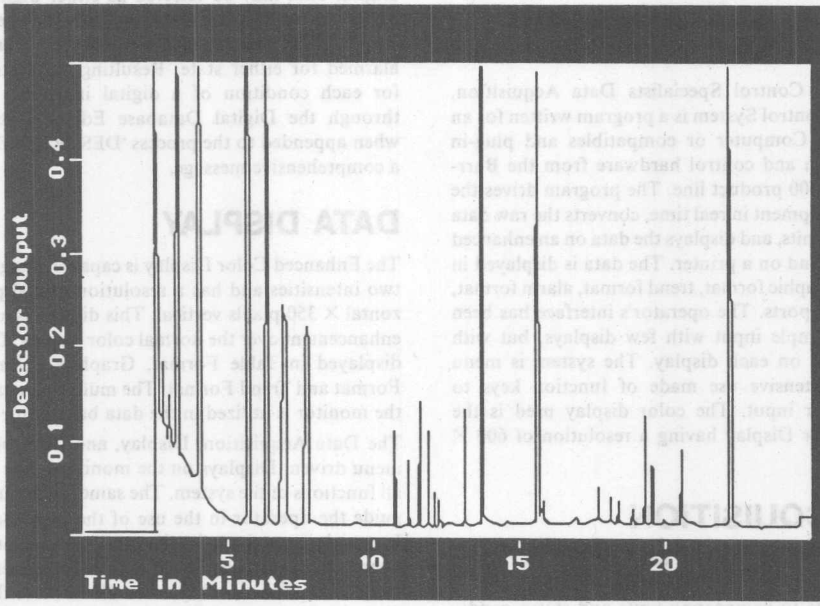
EASE OF USE

CHROM is extremely easy to learn and use. Its user interface is similar to LABTECH NOTEBOOK and Lotus 1-2-3. To set up the analysis method, the user is presented with a fill-in-the-blanks worksheet. This worksheet contains the existing method as a starting point. The method is modified very easily, by moving the cursor to the item to be changed, and typing in the new value. When all the changes have been made, the method is automatically saved to permanent disk storage, and becomes the new existing analysis method.

ANALYSIS METHOD PARAMETERS

The analysis method parameters include the following:

1. Maximum spike width and minimum spike amplitude
2. Number of points for smoothing/convolution
3. Automatic Slope Sensitivity (optional)
4. Maximum Peak Width, Minimum Peak Height, Maximum Baseline Width, Maximum Baseline Drift, Slope Check Width
5. Report Detail Level



LABTECH NOTEBOOK Scrolling Chromatograph Display.

LABTECH® Laboratory Technologies Corp.; IBM® International Business Machines Corp.; Lotus™, 1-2-3™, and Symphony™ Lotus Development Corp.; RS/1™ BBN Software Products Corp.

Software for Data Acquisition, Display and Control

Available from: **Industrial Control Specialists, Inc.** • P.O. Box 6471 • Lake Charles, LA 70605, USA
Telephone: 318-474-3163

FEATURES

- PCI-20000 compatibility
- Operates with IBM and compatible personal computers
- User friendly; no programming required
- Enhanced graphics display
- Historical trend display without compression
- Control functions: PID, On/Off, And, Or, Not, Switch
- Excellent graphics builder and database editor
- Alarm functions: Displays, Reports, Verification
- PID tune display
- Builds data files acceptable by Lotus 1-2-3
- Long-term archive of data with on-line recall
- Tailored to user's special functions
- Quantity discounts available

DESCRIPTION

The Industrial Control Specialists Data Acquisition, Display, and Control System is a program written for an IBM Personal Computer or compatibles and plug-in data acquisition and control hardware from the Burr-Brown PCI-20000 product line. The program drives the acquisition equipment in real time, converts the raw data to engineering units, and displays the data on an enhanced color monitor and on a printer. The data is displayed in table format, graphic format, trend format, alarm format, and in alarm reports. The operator's interface has been designed for simple input with few displays, but with maximum data on each display. The system is menu driven with extensive use made of function keys to reduce operator input. The color display used is the Enhanced Color Display having a resolution of 600 × 350 pixels.

DATA ACQUISITION

The Data Acquisition, Display, and Control System will acquire raw analog and digital data from the PCI-20000, convert the data to engineering units and status conditions, and display the data on an enhanced color monitor, and on a printer.

Configuration consists of defining the types of input/output modules installed on the carrier and parameters associated with each module. Once configured, the multiplexer will scan analog and digital data without stopping, and store the converted data in memory for access by the host computer.

CONVERSION OF ANALOG DATA TO ENGINEERING UNITS

Conversion routines are included for the following types of measured variables:

1. IC thermocouples (type J), degrees F
2. CA thermocouples (type K), degrees F
3. Differential liquid flow
4. Differential gas flow
5. Differential steam flow
6. Linear flow element (magnetic meter, vortex shedding, etc)
7. Linear input element (pressure, level, weight, analysis, etc)
8. Calculated variable.

CONVERSION OF DIGITAL DATA

Digital data may be selected as either digital input or digital output. Digital input data may then be associated with a normally open/closed process function and alarmed for either state. Resulting descriptive phrases for each condition of a digital input are assignable through the Digital Database Editor. These phrases when appended to the process 'DESCRIPTION' become a comprehensive message.

DATA DISPLAY

The Enhanced Color Display is capable of eight colors in two intensities and has a resolution of 600 pixels horizontal × 350 pixels vertical. This display is a significant enhancement over the normal color display. Data will be displayed in Table Format, Graphics Format, Alarm Format and Trend Format. The multi-color capability of the monitor is utilized in the data base editor also.

The Data Acquisition, Display, and Control System is menu driven. Displays on the monitor guide the user in all functions of the system. The same technique is used to guide the operator in the use of the Data Base Editor. Instructions are included in the displays that direct the operator in each function of the system. Extensive use is made of the function keys. The 25th line of the monitor display will always define the functions assigned to the ten special function keys. As displays change, the assignments of the function keys will change.

TABLE DISPLAY

The Table Display consists of 20 process variables in a tabular format. The process variables may consist of a mix of analog inputs, pulse inputs, digital inputs, digital outputs, analog outputs, and psuedo data (PID, Logical Functions).

Multiple table displays are available and a variable may appear in more than one display.

GRAPHIC DISPLAY

The process being monitored will be flow-charted on the monitor using the multicolor capability of the monitor. All variables associated with the flow chart being displayed will appear on the monitor in engineering units. Special characters have been designed for the monitor to allow for display of valves, pumps, heat exchangers, etc. in a significantly smaller space than that required by the ISA character set. Multiple graphic displays will be available. A variable may appear in more than one display.

TREND DISPLAY

Four process variables will be trended on two graticules (two per graticule). A trend line and its associated scale, description, and present value will be the same color. Data can be presented in three trend formats: rapid trend, present trend, or historic trend. Multiple trend displays are available.

ALARM DISPLAY

The Alarm Display consists of the latest 20 alarms with the latest alarm being at the top of the display.

HARDCOPY

Printers are used to produce a hardcopy of the status of alarms and to duplicate in black on white the table display or the trend display presented on the color monitor. On occurance of any alarm condition, the printer will print a comprehensive message containing the time, tag, service, present value, alarm limit, engineering units, and an indicator of limit violated.

CONTROL FUNCTIONS

Proportional/Integral/Derivative (PID) and Digital Output controls are included in the Data Acquisition, Display, and Control System. The PID function is implemented in a building block manner, where an analog input entry in the data base serves as an input to a PID control entry in the data base, which serves as an input to an analog output entry in the data base. Thus to implement PID control, three entries must be defined in the data base: two which have real signals connected and one (PID) that is a psuedo entry (calculated variable). If cascade control is required, an additional analog input entry and PID entry must be defined in the data base.

The Digital Control function provides for toggling digital outputs.

In addition to the standard provisions for digital input and digital output in the data base, entries in the digital database may be assigned to any of several logical functions (AND, OR, NOT, SWITCH).

SPECIFICATIONS

The Data Acquisition, Display, and Control System utilizes the following:

PCI-20000 Hardware

One General Purpose Carrier Board, model PCI-20001C-2

OR

- One High Performance Carrier Board, model PCI-20041C-3
- One Analog Input Module, model PCI-20002M-1, with eight differential inputs, 12-bit resolution, 25μsec conversion time
- One Analog Output Module, model PCI-20003M-2, with 2 outputs, 12-Bit resolution
- One Digital Input/Output Module, model PCI-20004M-1, with 32 channels of input or output, TTL signal levels
- One Counter/Timer/Pulse Generator Module, model PCI-20007M-1

COMPUTER

An IBM or compatible Personal Computer model XT or AT with the following features: 640 k byte RAM, 20 M byte fixed disk, 360 k byte floppy disk or 1.2 M byte floppy disk, keyboard, PC-DOS 2.1 or 3.1.

DISPLAY

IBM Enhanced Color Display, Feature 5154
IBM Enhanced Graphics Adapter, Feature 1200
IBM Graphics Memory Expansion Card, Feature 1201
IBM Graphics Memory Module Kit, Feature 1203

OR

Quadram Enhanced Color Display, Model Quadchrom
Quadram Enhanced Display Adapter, Model QuadEGA

PRINTER

The printer utilized requires the Centronics Parallel Port configuration. The printer may be an Epson FX85 or FX286, or equal.

ONSPEC
Software for Process Control

Available from: **Heuristics, Inc.** • 9723A Folsom Blvd., Suite 231 • Sacramento, CA 95827, USA
Telephone: 916-369-6606 • Telex: 4940010

DESCRIPTION

The ONSPEC family of process control software products runs on personal computers such as the IBM PC/XT/AT and true compatibles such as the AT&T 6300 and 6300 Plus, the COMPAQ Deskpro, HP Vectra 40 Megabyte, and the DEC MicroVAX II/VAXmate. It is modular in design to allow the user to select the features necessary for a specific application:

- Plant documentation
- Operating and target summaries
- Analog and digital data logs
- Shift summary: historical logs
- Operating changes
- Trend graphs
- Real-time displays
- Process status displays
- Output to instrumentation
- Batch controls
- Multiple users
- Up to six remote units
- Recipe loading
- Multitasking

ONSPEC Control Software is the master tool kit of the ONSPEC family. It combines the best available operating system, spreadsheet, editor and process control programs for design: preparation of diagrams, lists, specifications and reports; for analysis: modeling and simulation; for process control: displays, alarms, trends, and historical reporting.

Displays are created using a standard process control character set that replaces the foreign language character set in the IBM and is supplied on an EPROM (the EPROM is plugged into your color board). All of the variable information is stored in data tables using fill-in-the blank data entry.

These tools allow you to orchestrate and direct data acquisition, process control, and management.

ONSPEC EXTENDED MEMORY—Access up to 8Mb of RAM!

This ONSPEC Version breaks through the 640K RAM barrier. Using this version you can run more programs concurrently—such as ONSPEC with Supertrends, Control Blocks and Superintendent, as well as other DOS programs such as Lotus 1-2-3. This version also provides the power for additional remote terminal support.

ONSPEC Extended Memory requires an extended memory card such as the AST Rampage with a minimum of 2Mb of memory.

DEC MICROVAX II VERSION

The MicroVAX II/VAXmate Version of ONSPEC Control Software used the full capability of the computer to provide expanded monitoring and control capabilities—up to four times the capacity of an AT personal computer. This verison offers supervisory control possibility with a micro based system.

ONSPEC I/O INTERFACE—Conversing with the Real World

ONSPEC interfaces to the Burr-Brown PCI-20000 System of data acquisition and control hardware.

The I/O Template passes information between the PCI-20000 hardware and ONSPEC, allowing ONSPEC to monitor and control a process. The template is also responsible for creating and maintaining an internal scheduled list of commands to send.

The I/O Template configuration file defines the functions to be monitored, such as how often to poll a device variable, where to send the data, what kind of data conversion to use, or special commands applicable to the equipment.

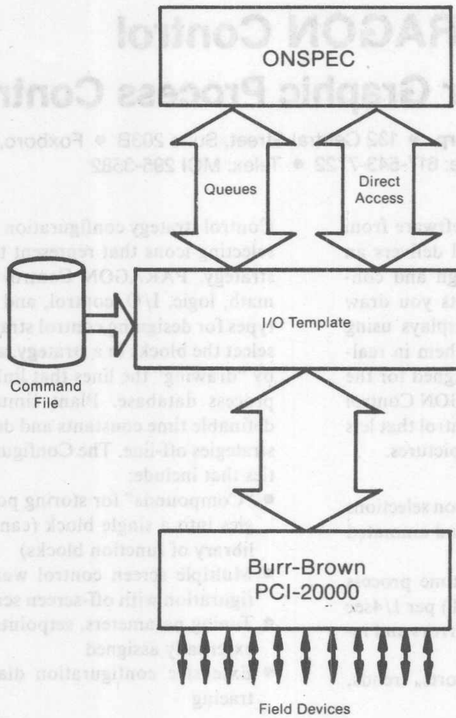
A sample configuration file including several different types of points comes with each I/O Template. A diagnostics program, provided with each template, assists in working with the configuration file and setting up the equipment.

Standard features include user definable conversions, variable template speed, COMM1 or COMM2 selectable, and the ability to run concurrently with many other ONSPEC templates.

The ONSPEC family of process control software products combined with personal computers and PCI-20000 hardware devices give the control engineer the freedom to develop designs with greater flexibility and lower cost than ever before. From design and simulation to supervisory control, we have products that will do the job: **ONSPEC Superintendent**—the real time expert system, **ONSPEC SQC** for statistical quality control, **ONSPEC Ladder Logic, OnNet** networking software, and **ONSPEC Supertrends**—the electronic implementation of the digital trend pen recorder are a few of the ONSPEC products that fully support the Burr-Brown PCI-20000.

ONSPEC SUPERINTENDENT

The real time expert system performs safety evaluations, reports and control procedures. ONSPEC Superintendent brings standard operating procedures on line and



can interact with process I/O and operators continually to make sure regulations are followed and exceptions documented.

ONSPEC SQC runs concurrently with ONSPEC control software and ONSPEC Superintendent to provide information on the degree of control for each process variable in a process. SQC limits can be established, referenced to alarm action, and stored in an historical file as a record of the process. Retrieved data can be manipulated through SQC calculations and operations to improve product quality.

ONSPEC Automatic Controller Tuning improves plant safety and operation. Menu driven ONSPEC ACT creates an optimal tuning algorithm and then calculates and tunes the parameters. ONSPEC ACT also tunes supervisory control algorithms supplied with ONSPEC Ladder Logic and ONSPEC Control Blocks.

ONSPEC COMMUNICATION OPTIONS

The exchange of information between people and computers, people and people, and computers and computers is essential in today's control environment. ONSPEC is designed with your multi-machine requirements in mind. We provide powerful tools for control engineers:

- ONSPEC Remote Support provides remote terminal color graphic display and editor capability to an ONSPEC supervisory control system.
- ONSPEC Remote Link adds four communication ports to an IBM XT/AT or fully PC-compatible computer. Devices can be connected to these ports so that up to four additional users can be supported simultaneously.
- ONNET is a communications program that allows two ONSPEC programs running on separate computers to exchange data.
- ONSPEC Network provides each computer on a networked multidrop the ability to access data from remote computers, run remote peripherals, and have full computing capabilities at each station, including virtual and remote consoles.

Call today for a complete catalog of the ONSPEC family of products.

PCI-20000 HARDWARE SUPPORTED

PCI-20001C-1 Carrier
PCI-20001C-2 Carrier
PCI-20002M-1 12-Bit 16-Channel DAS Module
PCI-200003M-2 12-Bit Two-Channel D/A Module

PARAGON Control

Software for Graphic Process Control

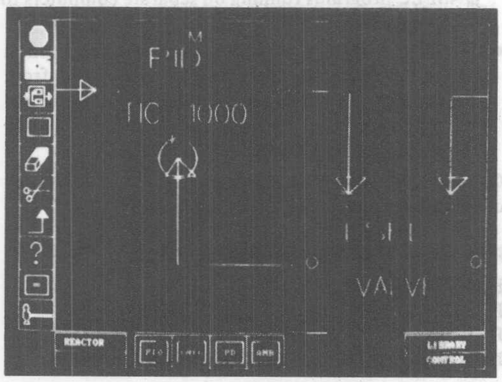
Available from: **INTEC Controls Corp.** • 132 Central Street, Suite 203B • Foxboro, MA 02035, USA
Telephone: 617-543-7722 • Telex: MCI 295-3582

The first in a series of packaged control software from Intec Controls Corp., PARAGON Control delivers an intuitive graphic approach to system design and configuration. Its icon-based user interface lets you draw control strategies and colorful, custom displays using predefined building blocks, and then run them in real-time without any programming effort. Designed for the IBM PC/XT/AT and industrial PC, PARAGON Control provides a visual environment for process control that lets users communicate with a process through pictures.

PARAGON Main Features Include:

- Mouse-driven Configurator module with icon selections for intuitive design of control strategies and animated operator graphics
- Interactive Runtime mode for fast real-time process control—30 loops (i.e., AIN, PID, AOUT) per 1/4sec
- EXPERT configuration checker reports errors and inconsistencies and recommends solutions
- Self-documenting control strategies, reports, trends, and process database records
- Process simulation feature allows testing and optimization of strategies in real-time without connecting to any I/O hardware
- Sequential execution of function blocks in the direction of signal flow
- Built-in I/O device drivers for Burr-Brown PCI-20000 Data Acquisition and Control System
- Historian with data exchange with Lotus 1-2-3/Symphony, RS/1
- On-line, context sensitive Help
- Fully documented with hands-on tutorial, indexed Reference Manual and Quick Reference Card.

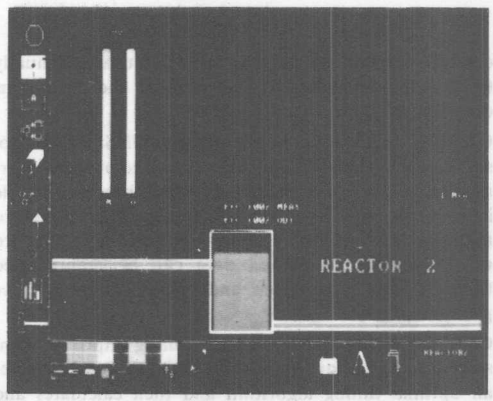
CONTROL STRATEGY CONFIGURATION



Control strategy configuration is achieved graphically by selecting icons that represent the building blocks of the strategy. PARAGON Control provides a full range of math, logic, I/O, control, and general expression block types for designing control strategies. A mouse is used to select the blocks in a strategy, and connections are made by "drawing" the lines that link selected elements in the process database. Plant simulation blocks with user definable time constants and delays let you run and tune strategies off-line. The Configurator has editing capabilities that include:

- "Compounds" for storing portions of complex strategies into a single block (can be appended to resident library of function blocks)
- Multiple screen control worksheet for strategy configuration with off-screen scrolling and pan and zoom
- Tuning parameters, setpoints, and logic inputs can be externally assigned
- Extensive configuration diagnostics and connection tracing

USER-DESIGNED OPERATOR GRAPHIC DISPLAYS

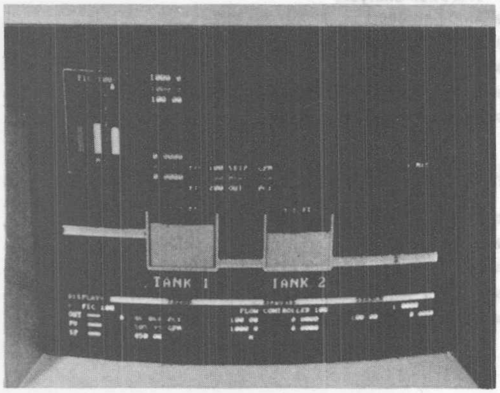


Users build their own operator graphic displays using an icon-based interface similar to the one for control configuration. Shapes, pre-built symbols and alphanumerics are either selected from the PARAGON library or defined by the user. Graphic symbols can be static or dynamic. Features in this mode of the Configurator include:

- Graphically connect symbols to the control strategy (no need to fill in menus)
- Cut and paste and replication of graphics symbols in any size

- Free-form "Pen" icon for user-created valves, motors, pumps, etc.
- User-created symbols can be saved in the library
- Dynamic selections include text size and color changes, or interactive data entry or display
- View up to 49 operator displays for real-time monitoring and control

REAL-TIME PROCESS MONITORING AND CONTROL



The Run module provides a convenient operating window on the plant. Two operating modes, one of which is password protected, allow process monitoring and control through user-selected displays. The split-screen interface lets you view the displays in a main window while simultaneously modifying tuning parameters in a block point display. Besides the custom displays, users can view a variety of built-in reports and real-time trends, all of which are bound to function keys or accessible through the command hierarchy. Among the features are:

- Built-in trends with variable time scales and selective point zoom
- Selective alarm activation and reporting, including static and dynamic alarm summaries
- Advanced reporting includes event log summaries and historian for recording selected data on disk (up to 100 points)
- Custom displays are linked in a display directory to allow quick "paging" through multiple operator graphic displays
- Block point displays to view and modify tuning values and other parameters
- Search-by-attribute for quick selection of data of interest
- Single key-stroke switching between operator displays, trends, alarm summaries and control strategies

APPLICATION

PARAGON's high quality operator graphics and runtime display strategies make it an efficient and cost-effective operator interface for process plant operations. Its unique combination of fast, flexible real-time control with secure distributed control applies equally to continuous or batch production. Smooth integration of external controllers with plug-in I/O cards and internal control algorithms gives you the power to mix and match the strengths of supervisory or direct digital control. Ideal for pilot plants and laboratory environments. PARAGON is equally suited for process diagnostics and loop tuning for large centralized processes. The simulation capability lets operators "get to know" and understand a plant under all operating conditions.

MINIMUM CONFIGURATION REQUIREMENTS

IBM PC/XT or PC/AT or hardware compatible
512kb memory RAM
10Mb hard disk
360kb disk drive
IBM EGA with 128K memory or compatible
IBM Enhanced Graphics Monitor or compatible
8087/80287 co-processor microchip
Mouse Systems Optical Mouse
MS-DOS 3.0 or higher
Most parallel printers (optional)

PCI-20000 HARDWARE SUPPORTED

PCI-20001C-2 Carrier
PCI-20002M-1 Module
PCI-20004M-1 Module
PCI-20005M-1 Module
PCI-20007M-1 Module
PCI-20021M-1 Module

Contact INTEC Controls for the latest list of PCI-20000 hardware supported.

SNAP-FFT

Software for Frequency Spectrum Analysis

Available from: **HEM Data Corp.** • 17025 Crescent Dr. • Southfield, MI 48076, USA
Telephone: 313-559-5607 • Telex: 467984 (Attn: HEM Data Corp.)

FEATURES

- Menu-driven, no programming required
- Converts time domain data to frequency domain
- Analyzes up to four channels of data
- Analyzes from 32 to 2048 points per channel
- Works in engineering units
- Reduces noise with spectrum smoothing
- Calculates transfer functions and impedances
- Calculates power spectra
- Plots results on linear or log scale
- Directs output to screen, printer, and/or disk

APPLICATIONS

- Vibration analysis
- Audio analysis
- Frequency response analysis
- Motion analysis
- Control system analysis
- Biomedical signal analysis

REQUIREMENTS (Minimum)

- IBM PC/XT/AT or compatible computer
- 384K memory
- IBM Graphics Adaptor or compatible
- Monitor with monochrome graphics capability
- DSDD floppy diskette or hard disk drive

DESCRIPTION

HEM Data Corporation's SNAP-FFT is a frequency spectrum analysis software package designed for the practical scientist and engineer. It converts the time domain data acquired with HEM's SNAPSHOT STORAGE SCOPE software* to the frequency domain using a Fast Fourier Transform (FFT) algorithm. Like SNAPSHOT STORAGE SCOPE, SNAP-FFT is menu-driven and requires no programming!

SNAP-FFT goes beyond the standard single or dual channel FFT and allows you to calculate the amplitude and phase for four channels of data. SNAP-FFT also takes the amplitude ratio of any two channels and the difference in the phase angle to determine the phase lag. These calculations provide the transfer function for control engineers, and the impedance value for electrical, mechanical, and acoustical engineers.

Advanced features, such as windowing, are available to modify the data if necessary. The user also has the option of selecting the portion of the time data to be analyzed in the frequency domain. The starting point can be selected along with the number of points to be analyzed. The number of points will be a power of 2: 32, 64, 128, 256, 512, 1024, 2048. For a single channel, up to 8192 points may be analyzed.

Menu selections are used to change the analysis parameters. The user can save menu settings and then recall them to quickly rerun various sets of data.

*SNAPSHOT STORAGE SCOPE is available from Burr-Brown as PCI-20068S-1. See Section 11.

Below are typical screens, taking data from SNAPSHOT Storage Scope (Figure 1) and converting them to the frequency domain. Figure 2 is a single channel FFT of channel 1. Figure 3 shows a transfer function for channels one and two from SNAPSHOT Storage Scope.

The data from SNAP-FFT can be in either tabular or graphical format. It can be displayed on the screen, recorded to disk, and/or sent to the printer. The graphs can be plotted on either a linear or log format.

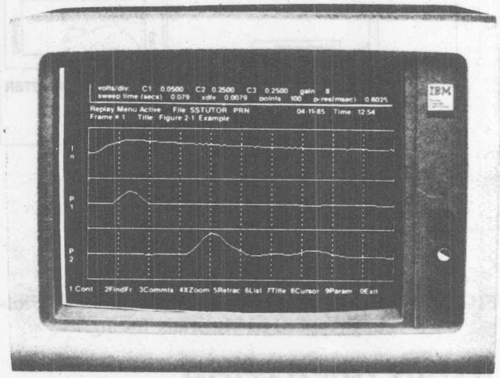


FIGURE 1. SNAPSHOT Storage Scope.

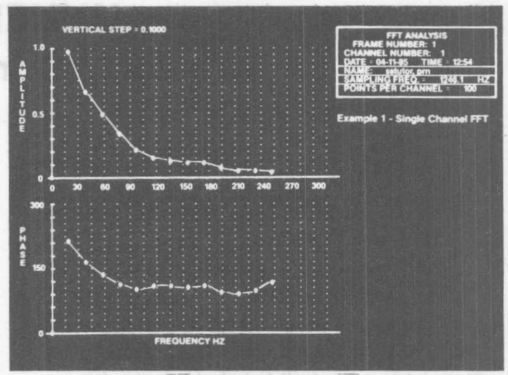


FIGURE 2. Single-Channel FFT.

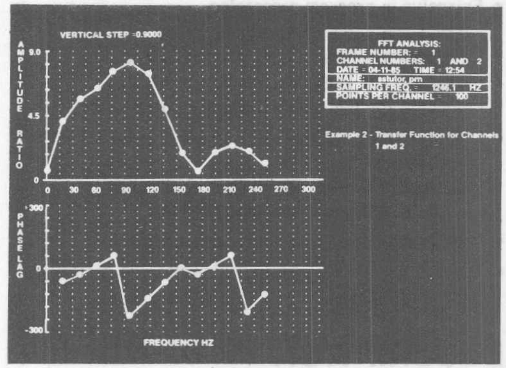


FIGURE 3. Transfer Function for Channels 1 and 2.

UNKELSCOPE

Software for Data Acquisition, Analysis, Display and Control

Available from: **Unkel Software, Inc.** • 62 Bridge St. • Lexington, MA 02173, USA • Telephone: 617-861-0181

FEATURES

- **Supports PCI-20000 Series Carriers and Modules for high speed data acquisition**
- **Completely menu driven software works with the IBM PC/XT/AT, and compatibles**
- **Display measured signals versus time or versus other measured signals in real time**
- **Store, retrieve data to disk, make hardcopy plots**
- **Extensive triggering modes and sources**
- **Extensive in-program processing:**
  - **–Transducer calibration and conversion**
  - **–Integration, differentiation, other functions**
  - **–High, low and bandpass filters**
  - **–Compute and plot spectral density, correlations**
- **Graphic editing of data:**
  - **–Zoom in on data**
  - **–Readout time, voltage from each of two cursors**
  - **–Compute enclosed areas, curve fit data**
- **Process control—proportional, PID, on/off**
- **Procedure (macro) capability automates and customizes to your applications**

DESCRIPTION

UnkelScope is an integrated software package for data acquisition, control, and data display, with extensive in-program data analysis capability. UnkelScope, compatible with the Burr-Brown PCI-20000 Series of data acquisition products, is a menu-driven interface that turns the IBM PC/XT/AT or compatible into a comprehensive replacement for a strip chart recorder, X-Y plotter, or oscilloscope. UnkelScope also adds extensive processing capability. UnkelScope requires no special computer instincts and is appropriate for scientists, engineers and technical support staff in a broad range of disciplines, with applications in research, development, teaching, production and testing.

The X versus Y feature of UnkelScope allows the display of cross-plots (as well as the normal signal versus time) in real-time. Previously taken data can be compared with currently sampled data. UnkelScope's graphical editing feature allows you to examine data in detail, finding values of particular points with a scrolling cursor, finding enclosed areas and fitting curves to sections of the data. The "macro" or procedure feature allows you to automate your measurement and analysis process. All of these functions and more are performed without programming.

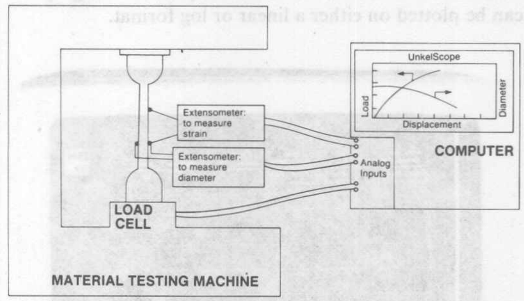


FIGURE 1. UnkelScope's Real-Time X Versus Y Plotting Applied to Materials Testing.

BASIC OPERATION

SETUP FOR DATA ACQUISITION

To specify a parameter with UnkelScope's compact, 'toggled' menu, you simply move the choice with the up and down arrow keys, and then 'toggle' to the desired value (shown in reverse video) using the left and right arrow keys.

The data sampling and display setup are shown in a single screen, so there is no confusion as happens with a layered menu. Figure 2 shows the setup screen, with the upper left region of the menu defining Vertical Trace 1 of the display. The input channel, trace label and the real-time display axis limits are selected here. For modules with software-programmable gain, the gain is selected by

```
Making Hardcopy of Setup Screen                               UnkelScope (c) MIT 1984
Commands          QUIT       SETUP MODE   SMPL/DSPLY   SAVE SETUP    GET SETUP
               SAVE DATA     PRNT SETUP   PRNT DSPLY    UTILITIES
 --------- Vertical Trace 1 ---------    -------- Additional Vertical Traces --------
| Source [Analog  0]  A/D Range [+/- 10] | Tr Input                               A/D |
| Label  :Voltage                        | #  Chan  Label:                      Range |
| Span   [  1    v full scale]           | 3 [none ]                                  |
| Range  [  .00E+00 to  1.00E+00]        | 4 [none ]                                  |
 ----------------------------------------| 5 [none ]                                  |
 --------- Vertical Trace 2 ---------    | 6 [none ]                                  |
| Source [None      ]                    | 7 [none ]                                  |
|                                        | 8 [none ]                                  |
|                                        |--------------------------------------------|
|                                        |  ------------- Triggering -------------    |
 ----------------------------------------| Mode  [Singl Sweep] Source [Keyboard ]     |
 --------- Horizontal Trace ---------    |                                            |
| Source [Time      ]                    |--------------------------------------------|
| Label  :Time (seconds)                 |  ------------- Processing -------------    |
| Span   [10   s full scale]              | Type [none                         ]       |
| Range  [  .00E+00 to  1.00E+01]        |                                            |
 ----------------------------------------|                                            |
 ------------- Sampling -------------    |                                            |
| Sample Rate [ 10 ms        100 hz]     |                                            |
| [1024] Samples  (Scan Time 1.02E+01 s) |                                            |
| (Delayed Plot  ) (Processing Inactive) |                                            |
```

FIGURE 2. Sample of the UnkelScope Setup Menu.

toggling to the desired value. In the example shown here, time has been chosen for the horizontal trace, but alternately any analog channel can be selected as the input to create real-time, Y versus X plots, a unique feature of UnkelScope. Sampling parameters and the comprehensive triggering options are also selected and displayed in the same menu.

DATA ACQUISITION, DISPLAY AND STORAGE

For low-speed signals, sampled data is plotted in real time on the computer monitor. For higher speeds, a full set of data is taken at the user-defined speed and is then plotted to the screen. Direct memory access (DMA) is used when available to achieve the highest possible sampling rates. A sample screen is shown in Figure 3. The upper trace is the raw data, while the lower trace shows the data after applying the low-pass filter option of UnkelScope. The data plot is well annotated with time, data, and trace labels. Sampled data with identifying time, date and label, can be saved in a disk file and then retrieved and replotted by UnkelScope.

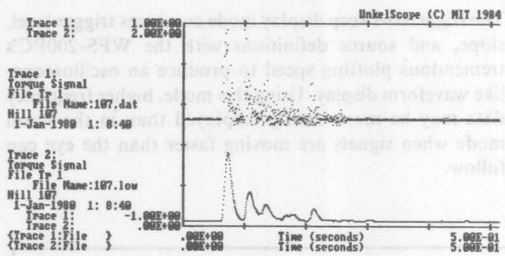


FIGURE 3. Sample of the Data Display Screen, Showing Raw Data and Low-Pass Filtered Data.

EXPERIMENT CONTROL AND PROCESS CONTROL

UnkelScope can also coordinate control of your experimental apparatus with data sampling or provide process control using a D/A output in an open-loop, proportional, PID or on/off controller.

IN-PROGRAM PROCESSING

UnkelScope's extensive in-program processing capability includes not only FFT related functions, but also digital filtering to improve signal quality, as well as integration, differentiation, and commonly used transformations. You can, for example, low-pass filter your data to remove high frequency noise, or determine the spectral content of a signal by computing and displaying the power spectral density.

GRAPHICAL EDITING

Two independent cursors can be moved through data with the numerical voltage and time values shown on the screen. You can also compute enclosed areas, isolate sections of the data, fit curves through sections of the data and perform other functions while seeing the data graphically.

PROCEDURES

An established sequence of data taking, processing and display can be automated using the procedure (macro) capability of UnkelScope. There is no new programming language to learn. Simply let UnkelScope remember your keystrokes as you take it through the procedure the first time.

DOCUMENTATION, SUPPORT AND UPDATES

UnkelScope comes with complete documentation, including a user's guide, an installation guide, a short guide and a tutorial disk. With your purchase of UnkelScope you will receive phone service and software updates for one year.

SPECIFICATIONS

Maximum channels	8 sampled, 2 displayed
Horizontal axis	Time or any analog input
Slowest sampling	Once each 500 seconds
Fastest sampling	Limited by hardware
Triggering options	Auto single sweep, with signal source from external, keyboard or analog input

MINIMUM SYSTEM REQUIREMENTS

- IBM PC, PC/XT, PC/AT, or fully compatible computer.
- Two diskette drives, or one hard disk and one diskette drive.
- UnkelScope is shipped with two versions:
  -256k version—allows 1024 points on each channel (uses the coprocessor if present but does not require it)
  -512k version—allows 4096 points on each channel and requires a math coprocessor.
- IBM Color Graphics Board (or compatible) or Hercules Graphics Card (or compatible)
- PCI-20000 Series System
  -PCI-20041C-2 or PCI-20041C-3 Carrier
  -PCI-20019M-1 A/D Module
  -PCI-20003M-2 D/A Module (required for process controllers

Inquire about support for other modules and carriers.

ORDERING GUIDE

UnkelScope Level 2+ (model number 2P-BB).

UnkelScope™ Massachusetts Institute of Technology.

WAVEFORM SCROLLER
Hardware and Software for Real-Time Waveform Plotting

Available from: **Dataq Instruments, Inc.** • 825 Sweitzer Ave. • Akron, OH 44311, USA
Telephone: 216-434-4284 • Telex: 650-281-7028

FEATURES

- **Orders of magnitude faster than bit-mapped graphics approaches.**
- **Fixed vector plot time of only 64μs for the IBM PC/AT yields the fastest waveform plotting available.**
- **Selectable smooth scroll and triggered sweep display modes allow bright, clear plotting of low and high frequency waveforms.**
- **Displays one to eight waveform channels.**
- **Directly accepts two's complement ADC values—drastically reduces software overhead.**
- **Ten predefined display formats for maximum display flexibility providing single or multiple display windows, and overlapping or non-overlapping traces.**
- **Relocatable software driver that adapts to any programming environment.**
- **Works in tandem with any existing video card or monitor.**

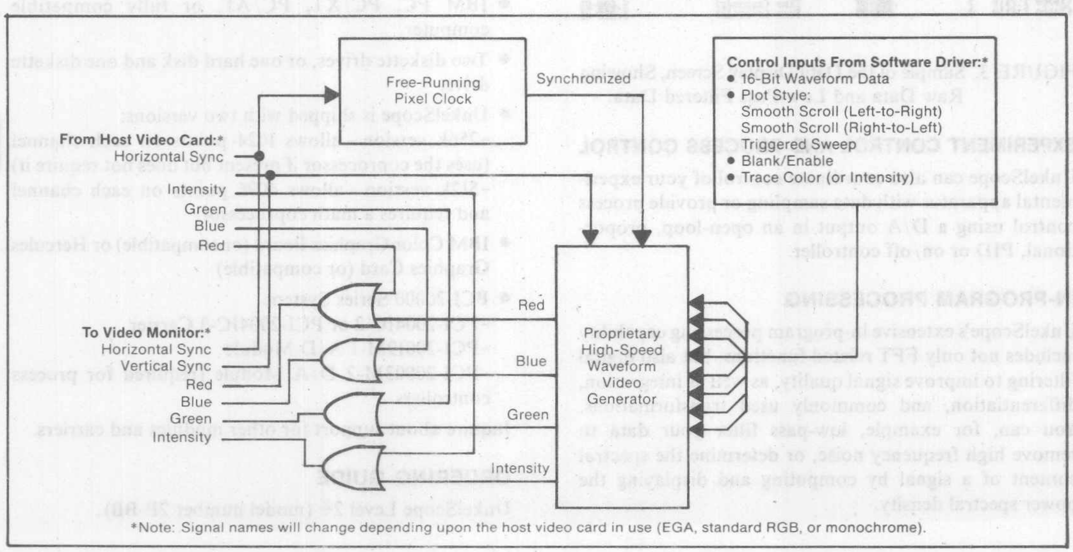
DESCRIPTION

The model WFS-200PC Waveform Scroller card breaks new ground in the quest for faster on-screen waveform graphics. Designed to overcome the speed limitations of waveform plotting through standard bit-mapped graphics cards, the WFS-200PC plug-in card applies an entirely different, hardware-intensive approach to waveform vector generation. The result is maximum display flexibility while minimizing software burden.

THE SCROLL AND TRIGGERED-SWEEP DISPLAY MODES

The WFS-200PC's scroll mode of operation displays digitized data as a smooth scrolling graph in chart recorder-like fashion. In a real-time waveform plotting sense, the scroll mode is especially useful for viewing lower frequency signals such as pressure, most biophysical potentials, temperature, electrical drift, and absorbance. The smooth scroll display mode provides the ultimate in waveform continuity since the real time focal point is always fixed, and an entire screen of historical waveform data is available at a glance.

The triggered-sweep display mode combines trigger level, slope, and source definitions with the WFS-200PC's tremendous plotting speed to produce an oscilloscope-like waveform display. Using this mode, higher frequency data may be more clearly displayed than in the scroll mode when signals are moving faster than the eye can follow.



WFS-200PC Block Diagram.

SPECIFICATIONS

Supported computers	IBM PC, PC/XT, PC/AT, and compatibles.
Supported video standard	Separate sync and video (noncomposite).
Supported host video cards	Any compatible with the above monitors (noncomposite only).
Pixel resolution (H×V)	Selectable 640×350, 640×200, 400×350, 400×200.
Video output impedance	Standard LS TTL load.
Vector draw time (any length)	64μs.*
Plot rate (throughput)	15,000Hz.*
Pixel draw rate	5,250,000 pixels per second.
Plot format	Continuous dot-joining.
Waveform data format	Two's complement 16-bit binary integer.
Number of waveform channels	One to eight max.
Waveform display modes	Continuous smooth scroll, triggered sweep.
Triggered sweep conditions:	
Trigger modes	Selectable auto/normal.
Trigger level (normal mode)	+32767 to −32768.
Trigger source	Any displayed channel.
Trigger slope	Positive/negative.
Overlapping waveform color/intensity:	
Monochrome	Even channels bright, odd channels dim.
Color	8-color selectable.
Power requirements	+5VDC at 1A.
Operating temperature range	0–50°C (32–122°F).
Operating humidity	20–90% noncondensing.
Expansion slot requirements	One long expansion slot.

*IBM PC/AT only. For PC and PC/XT: vector draw time is 250μs, plot rate is 4,000Hz.

Specifications subject to change without notice.

SOFTWARE SUPPORT FOR THE WFS-200PC

User-Supplied Software

Provided with the WFS-200PC is a software driver that may be accessed from a user-generated applications program to fully tap the product's waveform graphics power. The applications program may be written in any language such as BASIC, FORTRAN, C, Pascal, and assembly level. Dataq Instruments provides, with every WFS-200PC package, the software driver source code, a linkable object module for use with compiled applications programs, and a special binary-loadable software driver for use with interpreted BASIC programs. A programming manual complete with examples is provided.

Commercial Software Support

The WFS-200PC is supported by commercially available software packages such as LABTECH NOTEBOOK, available as Burr-Brown's PCI-20040S-1, described in this Handbook. Used with this software package, the WGS-200PC enhances real-time plotting capability by providing a continuous smooth-scroll or triggered-sweep display style, and extended real-time waveform plotting rates.

ORDERING INFORMATION

WFS-200PC—Waveform scroller for IBM™ PC series computers with monochrome or RGB monitors, including:

- Display format software providing a menu-level method of defining particular display organizations. Included are menu-level options for display format setup. Display mode selection (scroll or triggered sweep), and disk save and retrieve features.
- Run-time software providing all device-specific machine level routines used to service WFS-200 hardware. Included are software interfaces accessible through both high-level and assembly-level applications programs.
- All appropriate cable assemblies and adaptors.
- Complete documentation and limited one-year warranty. A complete warranty statement is provided with each order or upon request.

Attractive university, quantity, and OEM discounts are available. Contact Dataq Instruments or your local representative for more information.

For international orders, Dataq Instruments requires either prepayment or a confirmed, irrevocable letter of credit. All international orders are F.O.B. Akron, Ohio. Prices are in U.S. dollars and must include a 15% surcharge to cover additional shipping and handling.

NOTES

11

Section 12

GLOSSARY OF TERMS

This glossary consists of terms and their definitions as they normally pertain to the personal computer area and does not attempt to include every possible variation that might be used in the broader computer field that would include mainframes and specialized, dedicated machines.

active filter—An electronic filter that combines active circuit devices, usually amplifiers, with passive circuit elements, such as resistors and capacitors. Active filters typically have characteristics that more closely match ideal filters than do strictly passive filters.

algorithm—A set of rules or detailed plan, with a finite number of steps, for solving a problem. An algorithm can be used as a model for a computer program.

alias frequency—A false lower frequency component that appears in analog data reconstructed from original data acquired at an insufficient sampling rate.

analog-to-digital conversion (A/D)—The process of changing an analog signal into a digital value that represents the magnitude of the signal at the moment of conversion.

analog-to-digital converter (A/D or ADC)—An electronic device, often an integrated circuit, that produces a digital output directly proportional to an analog signal input.

ANSI—American National Standards Institute.

array—Data arranged in single or multidimensional rows and columns.

ASCII—American Standard Code for Information Interchange. A very popular standard method of encoding alphanumeric characters into 7 or 8 binary bits.

assembler—A program that converts a list of computer instructions written in a specific assembly-language format into binary instructions that can be executed by a specific processor.

assembly-language program—A program written directly with processor commands using mnemonic representations of the commands. The program is then processed by an assembler to produce executable machine code.

asynchronous—A communications protocol where information can be transmitted at an arbitrary, unsynchronized point in time, without synchronization to a reference timer or 'clock.'

background—Not visible to or not under the direct control of the computer operator.

bandpass filter—A type of filter that allows a band of signal frequencies between two set frequencies to pass while attenuating all signal frequencies outside the bandpass range.

base address—A memory address that serves as a point of reference. All other points are located by offsetting (adding to or subtracting from) in relation to the base address.

BASIC—The most common computer language, BASIC is an abbreviation for Beginners All-purpose Symbolic Instruction Code. Relies on English-like instructions which accounts for its popularity and ease of learning.

baud rate—Serial communications data transmission rate; the number of bits-per-second.

binary-coded decimal—A code for representing decimal digits in a binary format.

BIOS—Basic input/output system. Part of the computer's software operating system, BIOS is responsible for controlling data inputs from, and outputs to, peripherals such as the keyboard, screen display, printer, floppy disk and hard disk.

bipolar—A signal range that includes both positive and negative values.

bubble memory—A type of non-volatile computer memory that uses magnetic domains (bubbles) for data storage. Access to information stored in bubble memory is serial and therefore relatively slow compared to RAM. However, bubble memory is faster than floppy or hard disk. In addition, bubble memory is considerably more rugged than mechanical memory devices making it desirable in many industrial applications.

bus—Conductors used to interconnect individual circuitry in a computer. The conductors as a whole are called a bus.

byte—A term referring to eight related bits of information. Eight bits equals one byte.

C—A programming language, developed around the concept of structured programming, that bears a strong resemblance to PASCAL.

cache memory—Fast memory used to improve the performance of a CPU. Instructions that will soon be executed are placed in cache memory shortly before they are needed. This process speeds up the operation of the CPU.

CAD—Computer-aided design. A computer-based drafting and documentation system used for design engineering.

CAE—Computer-aided engineering. A computer system designed to aid engineering development.

CAM—Computer-aided manufacturing. A computer system used for automating manufacturing operations.

call—A software instruction used to pass control to a subroutine of a program. At the completion of this subroutine, control is returned to the original program at the point of the "call" statement. Often used for specialized routines such as "analog read" from a data acquisition system.

Carrier—In the PCI-20000 Personal Computer Interface System, the Carrier is the motherboard that "carries" a configuration of Instrument Modules to create a data acquisition/control system in a personal computer. The Carrier plugs into an expansion slot in an IBM PC, XT, AT or another compatible personal computer.

central processing unit (CPU)—The central part of a computer system that performs operations on data. In a personal computer the CPU is typically a single microprocessor integrated circuit.

code—As a noun, the text of a computer program. As a verb, to "code" means to write a program.

cold-junction compensation—A method of providing an artificial reference level and compensation for ambient temperature variations in thermocouple circuits.

command—An instruction given directly to an active program from a keyboard or terminal rather than from a program.

common-mode rejection ratio (CMR)—A measure of an instrument's ability to ignore or reject interference from a voltage common to its input terminals relative to ground. CMR is usually expressed in dB (decibels).

comparator—An electronic circuit used to compare two values and set an indicator that identifies which value is greater.

compiler—A particular type of high level language used to preprocess a program in order to convert it to a form that a processor can execute directly.

concurrent—software that can perform more than one task simultaneously.

Concurrent PC-DOS (C/PC-DOS)—Multitasking, multiuser disk operating system for the IBM PC family from Digital Research, Inc.

contact closure—The closing of a switch, often controlled by an electromagnetic or solid state relay.

conversion time—The time required, in an analog input or output system, from the moment a channel is interrogated (such as with a read instruction) to the moment that accurate data is available. This could include: switching time, settling time, acquisition time, A/D conversion time, etc.

coprocessor—Another computer processor unit that operates in conjunction with the standard CPU. Can be used to enhance execution speed. For example, the 8087 is designed to do floating point arithmetic.

counter—In software, a memory location used by a program for the purpose of counting certain occurrences. In hardware, a circuit that can count pulses.

cross assembler—A computer program that translates machine language code so that it can be read by a different type CPU.

crosstalk—In communications, a phenomenon in which a signal in one or more channels interferes with a signal or signals in other channels. In an analog multiplexer, the ratio of the output voltage to the input voltage with all channels connected in parallel and turned off.

current loop—Communications method that allows data to be transmitted over relatively long distances and through relatively high noise environments. With a current loop, the voltage levels are converted to currents so that the signals are transmitted in the form of current instead of voltage in a closed-loop circuit. Current loops are less sensitive to noise pickup.

data acquisition—Gathering information from sources such as sensors and transducers in an accurate, timely and organized manner. Modern systems convert this information to digital data which can be stored and processed by a computer.

data reduction—The process of analyzing a large volume of data to extract and refine a subset of the data for some particular purpose. As in the statistical summarization of data.

debouncing—Either a hardware circuit or a delay built into software to prevent false inputs from a bouncing key or switch contact.

decibel—A logarithmic measure of the ratio of two signal levels: $dB = 20Log_{10}(V_1/V_2) = 10Log_{10}(P_1/P_2)$.

default—A value assigned or an action taken automatically unless another is specified.

digital—A signal which has distinct states. Digital computers process data as binary information having either 1 or 0 states.

digital-to-analog conversion—The process of changing discrete data into a continuously varying signal. Common uses are to present the output of a digital computer as a graphic display or as a test stimulus.

digital-to-analog converter (DAC)—A device that converts digital information into a corresponding analog voltage or current.

DIP switch—A set of switches contained in a dual in-line package.

direct memory access (DMA)—A method by which information can be transferred from the computer memory to a device on the bus while the processor does something else. Also one of three methods of transferring data acquisition system measurements to computer memory, (the other methods being polling and interrupt).

DMA—Direct memory access, see above.

DOS—Disk operating system.

down-load—The copying of information from one computer to another.

drivers—Part of the software that is used to control a specific hardware device such as a data acquisition board or a printer.

DTE—Data terminal equipment.

duplex—The ability to both send and receive data simultaneously over the same communications line.

dynamic range—The ratio of the full scale range (FSR) of a data converter to the smallest difference it can resolve. Dynamic Range (DR) = 2^n. Generally expressed in dB, DR = 20 log 2^n. "n" is the resolution in bits.

event counter—A circuit used to count pulses that are related to the occurrences of a certain condition, such as an item coming off the end of the assembly line. An event counter can typically be preset, reset and can totalize.

expansion board—A plug-in circuit board that adds features or capabilities beyond those basic to a computer, such as a data acquisition system expansion board.

expansion chassis—An enclosure used to increase the capabilities of a computer system by providing space for additional expansion boards.

expansion slots—The spaces provided in a computer for expansion boards that enhance the basic operation of the computer.

expert systems—A highly specialized data base and computational computer program that acts like a human expert on a particular subject.

firmware—A program permanently recorded in a ROM and therefore essentially a piece of hardware that performs software functions. BIOS is an example of firmware.

floating-point numbers—Numbers that contain decimal parts or are presented in scientific notation (digits multiplied by a power of 10). Also known as "real" numbers. Integers are a subset of reals containing whole numbers only.

foreground—In a PC system, the activity subject to direct operator intervention. Other (background) activities continue as previously defined.

front end—The preprocessing of data before a program uses it. Could refer to signal conditioning in a data acquisition system.

GPIB—General-purpose interface bus. A standard bus used for controlling electronic instruments with a computer. Also designated IEEE-488.

ground—An electrically neutral wire having the same potential as the surrounding earth. Normally, a non-current carrying circuit intended for safety purposes. A reference point for an electrical system.

hardware—The visible parts of a computer system, such as the circuit boards, chassis, enclosures, peripherals, cables, etc. It does not include data or computer programs.

hexadecimal—A numbering system to the base 16.

hierarchical—A method of organizing data with a series of levels, each with further subdivisions, as in a pyramid or tree structure.

high-level language—A program used to simplify the creation of computer code. Allows the specification of a computer action using a smaller number of steps than assembly language.

IEEE-488—The Institute of Electrical and Electronic Engineers' designation for the GPIB instrumentation control bus standard.

I^3 Bus—Intelligent Instrumentation Interface Bus. A patent-pending interconnection bus for modular data acquisition components which is used to create a complete personal computer interface system.

input/output (I/O)—The process of transferring data from or to a computer system including communications channels, operator interface devices, or data acquisition and control channels.

Instrument Modules—In the PCI-20000 Personal Computer Interface System, the Instrument Modules provide specialized data acquisition functions. A configuration of Instrument Modules is combined with a motherboard Carrier to create a data acquisition/control system in a personal computer.

instrumentation amplifier (IA)—An amplifier circuit with both high-impedance differential inputs and high common-mode rejection.

integer—A whole number, not requiring a fraction, a decimal point or scientific notation for representation.

integrating A/D converter—An A/D conversion technique in which the analog input is integrated over time. Different types of integrating A/D converters include dual slope, triple slope, and charge balancing types.

interpreter—A high-level language in which the command statements are converted, one at a time and in the order they are used, into code that can be executed by the processor.

interrupt—A computer signal indicating that the CPU should suspend its current task to service a designated activity. One of three methods for transferring data acquisition measurements to the computer's memory, (the other methods being DMA and polling).

interrupt handler—The section of a program that performs the necessary operations to service an interrupt when it occurs.

I/O—Input/output, see above.

I/O address—A method that allows the CPU to distinguish between the different boards in a system. Unique addresses usually are set with DIP switches. All boards must have different addresses.

I/O mapping—Method of connecting I/O devices to the CPU in an addressable fashion without using memory space. Disk drives, printers and monitors are usually I/O-mapped. However, there is a limited address space available and a limited set of I/O instructions. For these reasons, advanced data acquisition systems tend to be memory-mapped.

isolation amplifier—An amplifier with electrically-isolated inputs and outputs which allows it to amplify a differential signal superimposed on a high common-mode voltage.

isolation voltage—The voltage which an isolated circuit can normally withstand. Isolation voltage is usually specified from input to input and/or from any input to the amplifier output, or to the computer bus.

isothermal—A process or area that is maintained at a constant temperature.

K—Kilo. In referring to computers, a "kilo" is 1024 or 2 to the 10th power. (Note that it is actually slightly more than an even 1000.)

latch—A term used to indicate that the state of a digital signal will remain stored until changed by the CPU or specified external command signal.

linearity—The adherence of a device's response to a straight line relationship.

linker—A program which combines different sections of a compiled program.

listener—A device on the GPIB bus that receives information from the bus.

machine language—Binary code that is executed directly by a computer CPU and translated into electronic actions. Machine language is different for each CPU type.

macro—A small set of program steps combined to act as a single, more powerful, program step.

memory—Electronic devices that enable a computer to store and recall information. In its broadest sense, memory refers to any hardware capable of serving that end, e.g., disk, tape, or semi-conductor storage.

mnemonics—A method of helping a software programmer remember the various commands of a specific computer system. A relatively easy-to-remember alphabetic code is assigned to each command and usually consists of letters extracted from, and thus suggestive of, the command it symbolizes.

modem—A device used to translate serial data to/from a form that can be transmitted/received over telephone or other communication channels. (Short for MOdulator-DEModulator.)

modular—The use of building blocks (modules) in a computer or data acquisition system. A modular device, for instance, is one that is built, tailored, and expanded by connecting various mutually compatible components. An example of a modular data acquisition system is the PCI-20000 Personal Computer Interface System.

monotonicity—The desirable characteristic of a digital-to-analog converter to produce a continuously increasing analog output for a correspondingly increased digital input code.

multidrop—A single communications line used to connect three or more points.

multiplexer (mux)—An array of semiconductor or electromechanical switches with a common output used for selecting one of a number of input signals.

multitasking—The characteristic of an operating system that allows a processor to perform several operations at once.

noise—An undesirable electrical interference to a signal. Sources of noise include the AC power line, motors, generators, transformers, fluorescent lights, soldering irons, CRT displays, computers, electrical storms, welders, radio transmitters, and others.

nonvolatile—A memory or data storage device that retains its information content when electrical power is removed. Ordinary RAM is volatile whereas ROM, bubble memory, battery-backed-up CMOS RAM, floppy, and hard disks are nonvolatile.

Nyquist Sampling Theorem—If a continuous bandwidth-limited signal contains no frequency components higher than a specified frequency then the original signal can be recovered without distortion if it is sampled at a rate of at least twice the specified frequency.

operating system—The master control program that governs the operation of a computer system. Software or firmware that manages the internal memory allocation and the control of peripheral devices for applications programs.

optical isolation—Two networks connected only through an optoelectric transmitter and receiver with no electrical continuity between the two networks.

overhead—The amount of computer processing resources, such as time or memory, required to accomplish some task.

passive filter—A filter circuit using only resistors, capacitors, and inductors. (No active devices such as integrated-circuit amplifiers.)

Pascal—A high-level programming language originally developed as a tool for teaching the concepts of structured programming. It has evolved into a powerful general-purpose language popular for writing scientific and business programs.

PCI—Personal computer instrumentation. Prefix for a series of personal computer interface products from the Burr-Brown Corporation and Intelligent Instrumentation Incorporated. For example, the PCI-20000 is a modular, board-level data acquisition and control system that fits directly inside a PC.

peripheral—The input/output and data storage devices attached to a computer such as disk drives, printers, keyboards, displays, data acquisition systems, etc.

PID—Proportional, integral, derivative. A three-mode control algorithm.

polling—A round-robin canvassing of data acquisition inputs synchronized in software to a clock or external trigger. One of three methods of transferring data acquisition measurements to the computer's memory, (the others being DMA and interrupt).

port—A communications connection on a computer or a remote controller.

protocol—The exact sequence of bits, characters and control codes used to transfer data between computers and peripherals through a communications channel.

quantizing error—The inherent uncertainty in digitizing an analog value due to the finite resolution of the conversion process. This error can be reduced only by increasing the resolution of the converter.

queue—A temporary storage location or list of things to be done such as messages that are awaiting transmission.

RAM—Random access memory, see below.

random access memory (RAM)—Computer memory

that allows data to be read or written at a particular location without having to pass sequentially through preceding locations.

range—Refers to the maximum allowable full-scale signal (input or output) that yields a specifed performance level.

rate generator—A device that provides a TTL-level pulse output at a software-programmable frequency.

read-only memory (ROM)—Computer memory in which data can be routinely read but written to only using special means when the ROM is manufactured. ROM is used for storing data or programs on a permanent basis.

real numbers—Numbers that can express a fractional value. Also called floating-point numbers.

real time—Data acted upon immediately instead of being accumulated and processed at a later time.

real-world—Referring to events, signals and conditions that occur naturally or in everyday life.

repeatability—The ability of an instrument to give the same output or reading under repeated identical conditions.

resolution—The smallest significant number to which a measurement can be determined. For example, a converter with 12-bit resolution can resolve 1 part in 4096.

ribbon cable—A flat cable in which the conductors are side by side rather than in a bundle.

ROM—Read only memory, see above.

routine—A self-contained program designed to accomplish a specific task.

RS-232C—A serial asynchronous communications standard used to connect modems, terminals and printers with serial interfaces. Although RS-232C is only specified for use in transmission lengths up to 50 feet, it is often used for greater distances at lower baud rates.

RTD—Resistance temperature detector. An electrical circuit element characterized by a positive coefficient of resistivity.

R/W—Read/write, abbreviation.

sample/hold—A circuit which acquires and stores an analog voltage on a capacitor for a short period of time.

sampling theorem—See Nyquist Sampling Theorem.

Seebeck effect—The basic principle behind thermocouples. When a circuit is created by the junctions of two dissimilar metals and the junctions are held at different temperatures, a current caused by the difference in temperature between the two junctions will flow in the circuit.

sensitivity—A measure of the minimum change in an input signal that an instrument can detect.

sensor—A device that responds to a physical stimulus (heat, light, sound, pressure, motion, etc) and produces a corresponding electrical output.

serial I/O—A common form of data transmission, in which the bits of each character are sent one at a time over the line.

serial port—A communications interface that uses one data line to transfer data bits sequentially. On the IBM PC the serial port refers to a standard serial interface which uses the RS-232C and ASCII standards.

set point—A "level" or control point in a feedback system.

settling time—The time required, after application of a step input signal, for the output voltage to settle and remain within a specified error band around the final value. The settling time of a system includes that of all of the components of the system.

shielded cable—A cable with foil or other sheathing around it to stop radio frequency interference and magnetic fields from generating extraneous signals on the cable conductors.

signal-to-noise ratio—On a communications line, the ratio of signal strength to the level of noise.

simultaneous sample/hold—A data acquisition system in which several sample/hold circuits are used to sample a number of analog channels at the same instant. One sample/hold per analog channel is required.

software—The non-physical parts of a computer system that includes computer programs such as the operating system, high-level languages, applications programs, etc.

span—The difference between the lower and upper limits of a range. Span is expressed in the same units as the range.

spike—A transient disturbance of an electrical circuit. Due, for example, to load variations on the AC power line.

stability—The ability of an instrument or sensor to maintain a consistent output when a constant input is applied.

strain relief—A bracket or clamp used to secure a cable so that it does not become disconnected accidentally or apply stress at the point of connection to the system.

subroutine—A sequence of computer instructions that perform a specific task and can be called repeatedly in a program whenever that specific task is required.

successive-approximation A/D converter—An analog-to-digital conversion method that sequentially compares a series of binary-weighted values with an analog input to produce an output digital word in "n" steps, where "n" is the bit resolution of the A/D converter. This process is analogous to weighing an unknown quantity on a balance scale using a set of binary standard weights.

surge—A sudden change (usually an increase) in the voltage on a power line. A surge is similar to a spike, but is longer lasting.

surge protector—A device placed in an electrical circuit to prevent spikes and some surges that might otherwise

damage electronic equipment connected to that circuit.

synchronization—The coordination of the activities of several circuit elements together.

syntax—Comparable to the grammar of a human language, syntax is the set of rules used for forming statements in a particular programming language.

20K—A slang term for the PCI-20000 Personal Computer Interface System.

talker—A device on the GPIB bus that simply sends information on to the bus without actually controlling the bus.

termination panel—A circuit board with screw terminations or another connector system that allows convenient connection of field signals to a data acquisition system.

throughput rate—The maximum repetitive rate at which a data conversion system can operate with a specified accuracy. It is determined by summing the various times required for each part of the system and then by taking the inverse of this time.

time stamp—Information added to a message, record, or other unit of data indicating the time at which it was processed by the system.

transducer—A device that converts length, position, temperature, pressure, level, etc, to a different energy form (i.e., voltage or current).

triac—A solid-state switching device used to switch alternating current waveforms.

turnkey—A system that combines all the hardware and software required for a specific application.

UART—universal asynchronous receiver-transmitter, see below.

universal asynchronous receiver-transmitter (UART)—An electronic circuit that translates the data format between a parallel representation within computer and the serial method of transmitting them over a communications line.

uninterruptible power supply (UPS)—A power conditioning unit placed between the commercial power service and the protected device. The UPS uses line power to charge batteries, which in the case of a power failure can drive electronic circuitry to produce the appropriate AC requirements for some time period.

UPS—Uninterruptible power supply, see above.

virtual memory—A method of making disk storage appear like RAM memory to the CPU, thus allowing programs to run that need more RAM memory than is installed in the system. This technique is slow compared to "real" memory.

volatile memory—Memory that does not retain its contents when power is removed.

voltage-to-frequency converter (VFC)—A device which converts an analog input voltage into a sequence of digital pulses with frequency proportional to the input voltage.

word—The standard number of bits that a processor or memory manipulates at one time. Microprocessors typically use 8 or 16-bit words.

Section 13

SUBJECT INDEX

NORTH AMERICAN SALES DIRECTORY

For Personal Computer Instrumentation Products

HOME OFFICE

Burr-Brown Corporation
International Airport Industrial Park
PO Box 11400
Tucson, AZ 85734-1400

Telephone 602-746-1111
TWX 910-952-1111
Telex 66-6491
Telefax 602-889-1510

BURR-BROWN OFFICES AND SALES REPRESENTATIVES

ALABAMA
Physimetrics
205-533-7591 Huntsville, AL

ALASKA
Burr-Brown Corp. (Home Office)
602-746-1111 Tucson, AZ

ARIZONA
Hannah-Hill Inc.
602-431-1771 Tempe, AZ

Burr-Brown Corp. (Home Office)
602-746-1111 Tucson, AZ

ARKANSAS
Acudata
405-743-1886 Stillwater, OK

CALIFORNIA (Northern)
Qualtech Associates, Inc.
415-680-1800 Concord, CA
408-732-4800 Sunnyvale, CA

CALIFORNIA (Southern)
Qualtech Associates, Inc.
714-639-9820 Orange, CA
805-495-7100 Thousand Oaks, CA

CALIFORNIA (San Diego County)
K-Rep Sales
619-457-9654 San Diego, CA

CANADA
Allan Crawford Associates
403-295-0822 Calgary, AB
Telex 03 821186
604-988-2195 N. Vancouver, BC
Telex 04 54247
Telefax 604-980-5892
416-890-2010 Mississauga, ON
Telex 06 961235
Telefax 416-890-1959
613-596-9300 Ottawa, ON
Telex 05 33600
514-747-7878 St. Laurent, PQ
Telex 05 824944

COLORADO
Hannah-Hill
303-460-0602 Broomfield, CO

CONNECTICUT
Technical Applications Associates, Inc.
203-327-2934 Stamford, CT

DELAWARE
Scientific Representatives Inc.
609-662-5222 Pennsauken, NJ

FLORIDA
Physimetrics
305-894-8081 Orlando, FL

GEORGIA
Physimetrics
404-955-0155 Atlanta, GA

HAWAII
Burr-Brown Corp. (Home Office)
602-746-1111 Tucson, AZ

IDAHO (Eastern)
Hannah-Hill
208-529-0671 Idaho Falls, ID
801-364-0606 W. Valley City, UT

IDAHO (Western)
M-Tech
206-854-6253 Kent, WA

ILLINOIS (Northern)
Computer and System Sales
312-830-7711 Hanover Park, IL

ILLINOIS (Southern)
Kebco Inc.
314-576-4111 St. Louis, MO

INDIANA
Wilson Technical Sales
317-872-2513 Indianapolis, IN

IOWA
Micro Resources Corporation
612-830-1454 Edina, MN

KANSAS
Kebco Inc.
913-541-8431 Overland Park, KS

KENTUCKY (Eastern)
Wilson Technical Sales
317-872-2513 Indianapolis, IN

KENTUCKY (Western)
Meta Technical Sales Inc.
216-562-8772 Aurora, OH

LOUISIANA
Acudata, Inc.
713-580-2451 Houston, TX

MAINE
Technical Applications Associates, Inc.
617-366-9781 Westborough, MA

MARYLAND
KD Associates
301-859-5151 Linthicum, MD

MASSACHUSETTS
Technical Applications Associates, Inc.
617-366-9781 Westborough, MA

MEXICO
Burr-Brown Corp. (Home Office)
602-746-1111 Tucson, AZ
TWX 910-952-1111
Telex 66-6491
Telefax 602-889-1510

MICHIGAN
Meta Technical Sales Inc.
216-562-8772 Aurora, OH

MINNESOTA
Micro Resources Corporation
612-830-1454 Edina, MN

MISSISSIPPI
Physimetrics, Inc.
205-533-7591 Huntsville, AL

MISSOURI
Kebco Inc.
314-576-4111 St. Louis, MO

MONTANA
Hannah-Hill
303-460-0602 Broomfield, CO

NEBRASKA
Kebco Inc.
913-541-8431 Overland Park, KS

NEVADA
Qualtech Associates, Inc.
415-680-1800 Concord, CA

NEW HAMPSHIRE
Technical Applications Associates, Inc.
617-366-9781 Westborough, MA

NEW JERSEY (Northern)
Gebhardt Marketing Inc.
201-376-4415 Springfield, NJ

NEW JERSEY (Southern)
Scientific Representatives Inc.
609-662-5222 Pennsauken, NJ

NEW MEXICO
Hannah-Hill
505-292-4098 Albuquerque, NM

NEW YORK (Metro Area)
Gebhardt Marketing, Inc.
516-582-3215 Long Island, NY
201-376-4415 Springfield, NJ

NEW YORK (Upstate)
Robert F. Lamb Co.
518-869-3355 Albany, NY
716-827-1111 Buffalo, NY
716-323-1250 Rochester, NY
315-455-2469 Syracuse, NY

NORTH CAROLINA
Jake Rudisill Associates, Inc.
704-377-6901 Charlotte, NC

NORTH DAKOTA
Micro Resources Corp.
612-830-1454 Edina, MN

OHIO
Meta Technical Sales Inc.
216-562-8772 Aurora, OH

OKLAHOMA
Acudata, Inc.
405-743-1886 Stillwater, OK

OREGON
M-Tech
503-639-7179 Tigard, OR

PENNSYLVANIA (Eastern)
Scientific Representatives Inc.
609-662-5222 Pennsauken, NJ

PENNSYLVANIA (Western)
Meta Technical Sales Inc.
216-562-8772 Aurora, OH

RHODE ISLAND
Technical Applications Associates, Inc.
203-327-2934 Stamford, CT

SOUTH CAROLINA
Jake Rudisill Associates, Inc.
704-377-6901 Charlotte, NC

SOUTH DAKOTA
Micro Resources Corp.
612-830-1454 Edina, MN

(continued on next page)